

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СКОРСЬКОГО»  
Факультет інформатики та обчислювальної техніки**  
(повна назва інституту/факультету)

**Автоматизованих систем обробки інформації і управління**  
(повна назва кафедри)

«На правах рукопису»  
УДК 004.04

До захисту допущено:  
В.о. завідувача кафедри  
\_\_\_\_\_ Олександр ПАВЛОВ  
«\_\_» \_\_\_\_\_ 2020р.

**Магістерська дисертація  
на здобуття ступеня магістра  
за освітньо-професійною програмою «Інженерія програмного забезпечення  
комп'ютеризованих систем»  
зі спеціальності 121 «Інженерія програмного забезпечення»  
на тему: «Програмне забезпечення попередньої обробки даних  
електрокардіограм»**

Виконала:  
студентка VI курсу, групи ІП-92мп  
Олійник Аліна Олексіївна

Науковий керівник:  
К.т.н., доцент кафедри АСОІУ  
Баклан Ігор Всеволодович

Рецензент:  
К.т.н., доцент кафедри АСОІУ  
Ткач Михайло Мартинович

Засвідчую, що у цій магістерській дисертації  
немає запозичень з праць інших авторів без  
відповідних посилань.  
Студентка \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Автоматизованих систем обробки інформації і управління**

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма - «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ Олександр ПАВЛОВ

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**

**Олійник Аліни Олексіївни**

1. Тема дисертації «Програмне забезпечення попередньої обробки даних електрокардіограм», науковий керівник дисертації Баклан Ігор Всеволодович, доц., к.н.д., затверджені наказом по університету від «26» жовтня 2020 р. № 3132-с
2. Термін подання студентом дисертації 15 грудня 2020 рік
3. Об'єкт дослідження процес попередньої обробки даних електрокардіограми
4. Вхідні дані файл з даними сигналу ЕКГ
5. Перелік завдань, які потрібно розробити аналіз існуючих підходів до обробки даних ЕКГ; удосконалити метод на основі лінгвістичного підходу; розробити програмне забезпечення, що реалізує удосконалений метод; провести експериментальні дослідження для виконання аналізу ефективності роботи запропонованого методу.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу діаграма компонентів; результати дослідження ефективності; загальна схема алгоритму перетворення даних в лінгвістичний ланцюг; схема алгоритму сегментації даних.

7. Орієнтовний перелік публікацій тези доповіді на науковій конференції ІСТУ-2020

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Графічний	доц. Ліщук К.І.		

9. Дата видачі завдання 12 жовтня 2020 рік

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Систематизація результатів огляду літератури	19 жовтня 2020	
2	Порівняльний аналіз існуючих методів розв'язання задачі	28 жовтня 2020	
3	Постановка та формалізація математичної моделі задачі	6 листопада 2020	
4	Модифікація існуючих методів розв'язання задачі	10 листопада 2020	
5	Розробка інформаційного та програмного забезпечення	18 листопада 2020	
6	Проведення експериментальних досліджень розробленого методу	20 листопада 2020	
7	Оформлення документації	24 листопада 2020	
8	Подання роботи на попередній захист	27 листопада 2020	
9	Подання роботи на основний захист	15 грудня 2020	

Студент

А.О. Олійник

Науковий керівник

І.В. Баклан

## РЕФЕРАТ

**Магістерська дисертація:** 98 с., 21 рис., 24 табл., 36 джерел, 2 додатки.

**Актуальність.** Хвороби, що пов'язані з роботою серця, є досить розповсюдженими. Необхідно вчасно діагностувати відхилення, щоб запобігти тяжких наслідків. Також слід враховувати, що зазвичай аналіз даних пацієнта проводиться людиною, що також може впливати на результат діагностування. Тому проблема попередньої обробки даних є досить актуальною для подальшого використання в експертно-діагностичних системах та програмних забезпеченнях.

**Зв'язок роботи з науковими програмами планами темами.** Робота

виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Методи та технології високопродуктивних обчислень та обробки надвеликих масивів даних». Державний реєстраційний номер 0117U000924.

**Мета дослідження:** розширення можливостей аналізу даних електрокардіограм за рахунок використання лінгвістичної моделі на основі апарату нечітких множин.

Для досягнення мети необхідно виконати наступні **задачі**:

- зробити огляд відомих рішень поставленої задачі в рамках дослідження;
- створити функцію зчитування даних та розбиття на такти сигнал ЕКГ;
- удосконалити метод перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання функції приналежності та правил нечітких множин;
- створити програмний модуль для реалізації удосконаленого методу;
- створити програмний модуль для візуалізації та збереження результатів обробки;
- дослідити ефективність удосконаленого методу.

**Об'єкт дослідження:** процес попередньої обробки даних електрокардіограм.

**Предмет дослідження:** методи попередньої обробки даних електрокардіограм.

**Методи дослідження,** що були використанні в даній магістерській роботі, базуються на методі лінгвістичного моделювання та використання апарату нечітких множин.

**Наукова новизна:** удосконалено метод перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання функції приналежності та правил нечітких множин. Таке розширення дозволяє застосувати апарат нечітких множин для подальшого аналізу даних пацієнта.

**Практичне значення отриманих результатів** визначається тим, що запропонований метод видає сигнал електрокардіограми у вигляді лінгвістичного ланцюга для подальшої класифікації та виявлення аномалій.

**Апробація:** Основні положення роботи доповідались і обговорювались на IV Всеукраїнській науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) у рамках доповіді на тему «Використання лінгвістичної моделі для попередньої обробки даних ЕКГ».

**Публікації.** Олійник А.О., Баклан І.В. Використання лінгвістичної моделі для попередньої обробки даних ЕКГ / А.О. Олійник, І.В. Баклан / IV Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р.

МЕТОД ЛІНГВІСТИЧНОГО МОДЕЛЮВАННЯ, ЛІНГВІСТИЧНИЙ ЛАНЦЮГ, ЕКГ, НЕЧІТКІ МНОЖИНИ, ПОПЕРЕДНЯ ОБРОБКА ДАНИХ

## ABSTRACT

**Master's dissertation:** 98 p., 21 pic., 24 tables, 36 sources, 2 appendixes.

**Topicality.** Heart diseases are quite common. Deviations must be diagnosed in time to prevent serious consequences. It should also be borne in mind that usually the analysis of patient data is performed by a person, which can also affect the outcome of the diagnosis. Therefore, the problem of data processing is quite relevant for further use in expert diagnostic systems and software.

**Relationship of work with scientific programs, plans, themes.** Work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute» within the topic «Methods and Technologies for High-Performance Computing and Processing of Large Data Sources». State Registration Number 0117U000924 .

**Aim of the research:** enhance the capacity of analysis of electrocardiogram data using a linguistic model based on the fuzzy sets.

**Task of the research:**

- review existing solutions to the problem in the study;
- create a function for reading data and dividing the ECG signal into segments;
- improve the method of converting the ECG signal into a linguistic chain by

adding the membership function and the rules of fuzzy sets;

- create a software to implement an advanced method;
- create a software module for visualization and storage of processed results;
- investigate the effectiveness of the improved method.

**Object of the research:** preprocessing of electrocardiogram data.

**Subject of the research:** methods of preprocessing of electrocardiogram data.

**Methods of the research:** based on the method of linguistic modeling and the fuzzy sets.

**Scientific novelty of the obtained result:** the method of converting the ECG signal into a linguistic chain by adding the membership function and the rules of fuzzy sets has

been improved. This extension allows the use of fuzzy sets for further analysis of patient data.

**Practical value of the results:** the proposed method produces an electrocardiogram signal in the form of a linguistic circuit for further classification and detection of anomalies.

**Link to scientific programs:** The main points of the work were reported and discussed at the Fourth All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" (ISTU2020) as part of a report on the topic "Using a linguistic model for pre-processing ECG data".

**Publications.** Oliinyk A. O., Baklan I.V. Using a linguistic model for pre-processing ECG data / A. O. Oliinyk, I.V. Baklan // Proceedings of the Fourth All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" (ISTU2020) - Kyiv: NTUU "KPI them. Igor Sikorsky", November 26-27, 2020.

LINGUISTIC MODEL, LINGUISTIC CHAIN, ECG, FUZZY SETS, DATA PREPROCESSING

## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ.....</b>	<b>10</b>
<b>ВСТУП .....</b>	<b>11</b>
<b>1 МЕТОДИ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ .....</b>	<b>14</b>
1.1 ОПИС ДАНИХ ЕЛЕКТРОКАРДІОГРАМ.....	14
1.2 ПРОЦЕС ОЦИФРУВАННЯ ДАНИХ .....	16
1.3 МЕТОДИ ОБРОБКИ ДАНИХ ЕЛЕКТРОКАРДІОГРАМ.....	19
1.3.1 Перетворення у вибірку .....	19
1.3.2 Перетворення Фур'є .....	22
1.3.3 Вейвлет-перетворення.....	22
1.3.4 Методи на основі градієнту та морфології.....	24
1.4 ВИКОРИСТАННЯ ЛІНГВІСТИЧНОГО МОДЕЛЮВАННЯ .....	25
1.5 ПОСТАНОВКА ЗАВДАННЯ.....	28
Висновки до розділу .....	29
<b>2 ЗАСТОСУВАННЯ АПАРАТУ НЕЧІТКИХ МНОЖИН.....</b>	<b>30</b>
2.1 ВИЗНАЧЕННЯ ПОНЯТТЯ НЕЧІТКІ МНОЖИНИ .....	30
2.2 ТИПИ ФУНКЦІЙ ПРИНАЛЕЖНОСТІ .....	31
2.2.1 Трикутна функція.....	32
2.2.2 Трапецієвидна функція .....	33
2.2.3 S-подібна функція.....	34
2.2.4 Z-подібна функція.....	35
2.3 ВИКОРИСТАННЯ ФУНКЦІЇ ПРИНАЛЕЖНОСТІ НА ОСНОВІ ЧАСТОТ.....	36
2.4 СИСТЕМА НЕЧІТКОЇ ЛОГІКИ .....	37
2.5 УДОСКОНАЛЕНИЙ МЕТОД НА ОСНОВІ АПАРАТУ НЕЧІТКИХ МНОЖИН .....	39
Висновки до розділу .....	40
<b>3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РЕЗУЛЬТАТИ.....</b>	<b>42</b>
3.1 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	42
3.2 ЗАСОБИ РОЗРОБКИ .....	43
3.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	44
3.4 ОПИС ЕКСПЕРИМЕНТУ .....	46
3.4.1 Опис вхідних даних .....	46
3.4.2 Опис вихідних даних .....	48
3.5 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ .....	51
Висновки до розділу .....	53
<b>4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ .....</b>	<b>55</b>
4.1 ОПИС ІДЕЇ.....	55
4.2 ТЕХНОЛОГІЧНИЙ АУДИТ ІДЕЇ ПРОЕКТУ .....	57



4.3	АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ ЗАПУСКУ СТАРТАП-ПРОЕКТУ .....	58
4.4	РОЗРОБЛЕННЯ РИНКОВОЇ СТРАТЕГІЇ ПРОЕКТУ .....	67
4.5	РОЗРОБЛЕННЯ МАРКЕТИНГОВОЇ ПРОГРАМИ СТАРТАП-ПРОЕКТУ .....	71
	Висновки до розділу .....	74
	<b>ВИСНОВКИ .....</b>	<b>75</b>
	<b>ПЕРЕЛІК ПОСИЛАНЬ .....</b>	<b>77</b>
	<b>ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ.....</b>	<b>81</b>
	<b>ДОДАТОК Б ПРОГРАМНИЙ КОД .....</b>	<b>86</b>

## ПЕРЕЛІК СКОРОЧЕНЬ

ЕКГ – електрокардіограма – запис змін електричних потенціалів серця.

QRS комплекс – комплекс, що містить інформацію про Q, R та S піки електрокардіограми.

Оцифрування – процес трансформації даних в електронний вигляд.

ПЗ – програмне забезпечення.

RTF – формат електронних документів, які створюються, редагуються та відкриваються за допомогою програми Microsoft Word.

JSON (JavaScript Object Notation) – текстовий формат, що дозволяє описувати об'єкти та інші структури даних для обміну інформації.

GUI (Graphical user interface) – набір засобів для взаємодії користувача з комп'ютерною програмною.

## ВСТУП

Аритмія серця є найпоширенішою та значною причиною захворюваності та смертності серед серцевих захворювань. Це група станів, які можуть характеризуватися порушеннями серцевого ритму. Рання діагностика має вирішальне значення для забезпечення вчасного лікування пацієнтів, які страждають цією хворобою. Традиційно діагностика проводиться шляхом огляду електрокардіограми (ЕКГ) кардіологом.

Електрокардіографія – метод графічної реєстрації змін різниці потенціалів серця, що виникають протягом процесів збудження міокарда. Перша реєстрація електрокардіосигналу, прототипу сучасної ЕКГ, була зроблена В. Ейнтховеном в 1912 р в Кембриджі. Після цього методика реєстрації інтенсивно вдосконалювалася. Сучасні електрокардіографи дозволяють здійснити як одноканальний, так і багатоканальну запис ЕКГ. Сьогодні в клінічній практиці найбільш широко використовують 12 відведень ЕКГ, запис яких є обов'язковою при кожному електрокардіографічному обстеженні хворого: 3 стандартних відведення, 3 посилені однополюсних відведення від кінцівок і 6 грудних відведень [1].

Процес запису електрокардіограми проводиться в спеціальному приміщенні, віддаленому від можливих джерел електричних перешкод: електромоторів, фізіотерапевтичних і рентгенівських кабінетів, розподільних електрощитів. До пацієнта прикріплюють електроди з манжетами. Для поліпшення якості даних і зменшення кількості наведених струмів слід забезпечити хороший контакт електродів з шкірою. До кожного електроду, встановленому на кінцівках або на поверхні грудної клітки, приєднують провід, що йде від електрокардіографа і маркований певним кольором. Запис даних проводять при спокійному диханні, а також на висоті вдиху. Пристрій оброблює імпульси серця та перетворює їх графічну криву.

У відділенні інтенсивної терапії монітори піднімають тривогу, коли вони вважають, що пацієнти перебувають у групі ризику. На жаль, часто виявляється, що ці монітори генерують помилкові тривоги. Дослідження виявило, що 88,8%

анотованих аритмічних тривог були помилково спрацьованими. Основними причинами було виявлено шум та артефакти у фізіологічних сигналах, що знову ж таки ймовірно були спричинені рухом пацієнта або розхитанням електродів. Занадто багато помилок може створити шумне середовище та спричинити стан серед медичного персоналу, що вони можуть ігнорувати справжні тривоги [2] Якщо використовувати алгоритми, що будуть видаляти шум та правильно відрізняти сигнали руху або розхитання електродів від справжніх відхилень роботи серця, це значно зменшить помилкові тривоги.

Найпоширенішим інструментом, що використовується для діагностики серцевої аритмії, є електрокардіограма (ЕКГ). ЕКГ - проста процедура для оцінки електричної активності серця. Інтерпретація ЕКГ відіграє центральну роль в оцінці серцево-судинних захворювань на основі досвіду кардіолога або комп'ютерних систем діагностики. На практиці автоматизована інтерпретація стає все більш важливою, оскільки така техніка покращує точність діагностики, полегшує процес прийняття рішень у галузі охорони здоров'я та зменшує витрати. Стандартний підхід полягав у проведенні оффлайн обробки сигналів ЕКГ, отриманих з онлайн бази даних, з використанням алгоритмів обробки сигналів, які виділяють потенційні ділянки аномалії для огляду ЕКГ-техніком або лікарем [1].

Зазвичай аналіз ЕКГ проводиться лікарем. Він переглядає отримані результати і, якщо знаходить хоч мінімальні відхилення, то направляє пацієнта на детальне обстеження. Проте не слід забувати, що навіть досвідчений фахівець може пропустити деякі важливі фактори та неправильно діагностувати. Для запобігання цього все більше автоматизується процес отримання, обробки та аналізу даних. Це не означає, що необхідно замінити лікарів, проте автоматизація значно зменшить навантаження та допоможе в діагностуванні.

З появою систем автоматичного аналізу необхідна перевірка якості сигналу, що знімаються без присутності лікаря. Якщо дані будуть передаватися в подальший аналіз належності якості, то система буде діагностувати з найменшим відсотком похибки. Для цього на сам перед дані від електрокардіограма оцифровуються. Видаляється шум, рамки, розмірна сітка, виконується фільтрація. Застосовуються

методи трансформації Фур'є або вейвлет-трансформації, щоб збільшити точність та деталізувати дані. На виході отримують інформацію у вигляді ряду.

Вище описані фактори обґрунтовують необхідність виділяти попередньої обробки сигналів великої уваги, оскільки від цього залежить результат виконання подальшої обробки даних. Правильність виконання роботи системи діагностування залежить не тільки від алгоритму аналізу, а й від вхідних даних.

# 1 МЕТОДИ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ

## 1.1 Опис даних електрокардіограм

Електрокардіограма (ЕКГ) - це квазіперіодичний, ритмічно повторюваний біомедичний сигнал, що включає інформацію про електричну діяльність серцевого м'яза. Сучасні електрокардіографи дозволяють здійснити як одноканальний, так і багатоканальний запис ЕКГ. В останньому випадку синхронно реєструються кілька різних електрокардіографічних каналів, що дає можливість отримати більш точну інформацію про електричне поле серця. Стандартна ЕКГ має 12 каналів, включаючи 6 каналів кінцівок (I, II, III, aVR, aVL, aVF) та 6 грудних каналів (V1, V2, V3, V4, V5, V6), записані від електродів на поверхні тіла. Точна інтерпретація ЕКГ для пацієнта з одночасними порушеннями серцевого ритму є складною навіть для досвідченого кардіолога, і неправильно інтерпретована ЕКГ може призвести до невідповідних клінічних рішень або призвести до несприятливих результатів [2].

Один цикл ЕКГ, який називається серцебиттям, окреслюється розташуванням сигналів P, QRS-комплексу та T, а також сегментами PQ та ST. З усіх форм хвиль QRS-комплекс є найбільш вражаючою формою сигналу, оскільки він представляє деполяризацію шлуночків і відображає основну частину електричної активності серця. Точне визначення розташування R-піку як найвищого та єдиного значимого піку в комплексі QRS відіграє вирішальну роль у отриманні морфології комплексу QRS. Крім того, локалізація піку R служить основою для автоматизованого визначення частоти серцевих скорочень, а отже, це важливий критерій для виявлення будь-яких серцевих аномалій. Приклад одного циклу електрокардіограми здорової людини зображено на рисунку 1.1.

Різні сегменти ЕКГ, такі як інтервал PR, інтервал QRS, інтервал QT, інтервал ST, сегмент PR та сегмент ST, використовуються для висновку про стан серця. Виявлення QRS діє як вихідна точка для вилучення інших функцій ЕКГ, таких як P, T-хвиля сегмента ST. Існують різні доступні алгоритми виявлення QRS, які загалом класифікуються як амплітудні та похідні, цифрові фільтри, методи узгодження шаблонів, нелінійні перетворення.

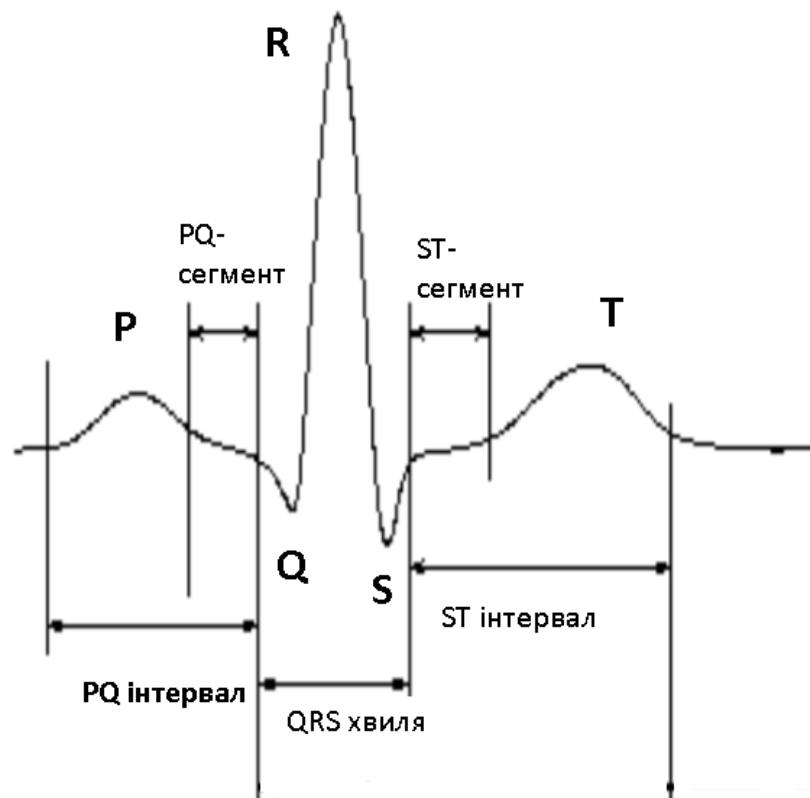


Рисунок 1.1 – Цикл серцевого ритму у здорової людини

Кожна ЕКГ включає різновиди кожної хвилі. Наприклад, комплекс QRS може мати різну форму, як показано на рисунку 1.2. Отже, сигнал ЕКГ є послідовністю серцебиття, що є аналогом речення в природній мові, і кожне серцебиття складається з безлічі хвиль, що схожу на слова в реченні [3].

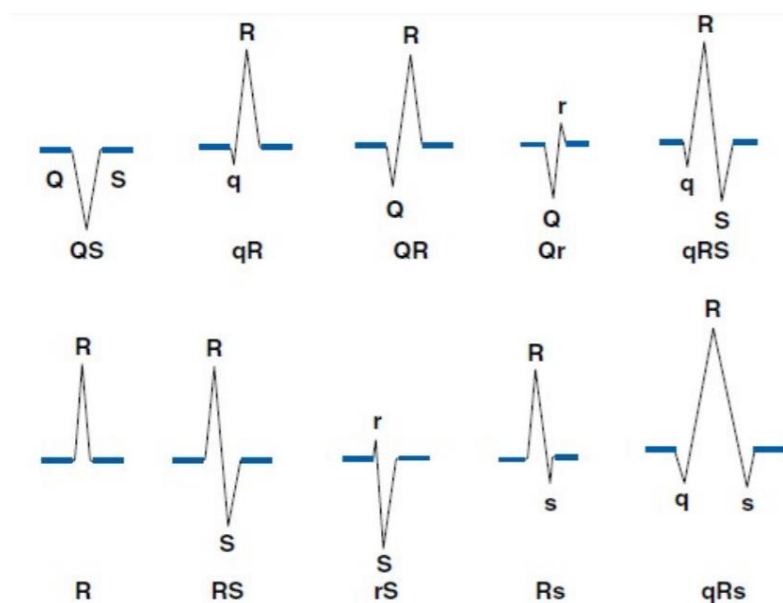


Рисунок 1.2 – Приклади комплексу QRS.[4]

Аналіз ритму серця включає визначення регулярності та частоти серцевих скорочень, джерела збудження, а також оцінку функції провідності. Регулярність серцевих скорочень оцінюється при порівнянні тривалості інтервалів R-R між послідовно зареєстрованими серцевими циклами. Інтервал R-R зазвичай вимірюється між вершинами зубців R. Регулярний або правильний ритм серця діагностується в тому випадку, коли тривалість інтервалів R-R приблизно однакова і розкид отриманих величин не перевищує  $\pm 10\%$  від 16 середньої тривалості R-R інтервалів. В інших випадках діагностується неправильний (нерегулярний) серцевий ритм [5].

## 1.2 Процес оцифрування даних

Для багатьох лікарень процес оцифрування ЕКГ-сигналів не проводиться, оскільки вони покладаються на друковану версію через високі витрати на модернізацію обладнання. Для зберігання даних про пацієнтів потрібно багато місця, особливо в дуже густонаселених регіонах, наприклад, в Індії чи Китаї. В такому випадку не можна гарантувати швидкий доступ до медичних карток пацієнтів.

Саме тому потрібне оцифрування. З одного боку, тривале зберігання записаних ЕКГ не призведе до втрати інформації. Плюс, швидка передача даних була б можливою при передачі інформації від одного медичного інституту до іншого. З іншого боку, для дослідження записаних сигналів можуть застосовуватися сучасні обчислювальні методи. Оцифрування ЕКГ також призведе до значного збільшення кількості даних, доступних для дослідницьких цілей, оскільки старі дані можна відновити, а нові додатково записати. Це стане особливо актуальним при роботі з рідкісними захворюваннями серця, для яких існує не так багато відомих випадків, але записи могли збиратися протягом багатьох років у минулому.

Для початку, записана ЕКГ сканується і застосовуються можливі методи обертання та перекошу. Приклад відсканованих даних зображено на рисунку 1.2.1. Потім виконується видалення фонові сітки. Для відновлення сигналу



застосовуються різні види технологій обробки. Наступним кроком є вилучення сигналу. Використовується розпізнавання хвиль, виділяють лінії сигналу. Час від часу додатково виконують оптичне розпізнавання символів для отримання цінної інформації про пацієнта.

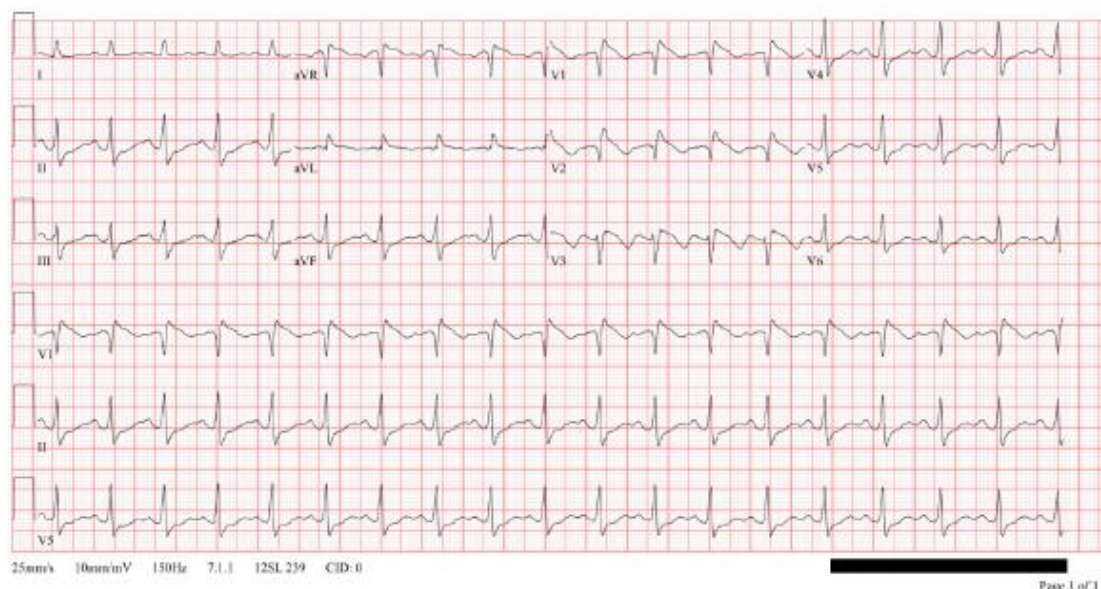


Рисунок 1.3– Скан ЕКГ

Якщо скан містить всередині чорну рамку або будь-що, що схоже на межі з додатковою, проте не потрібною інформацією, то необхідно це обрізати. Даний етап виконується для очистити дані від сторонньої інформації. Тому наступним кроком є отримання внутрішнього образу.

Далі проводиться операція порогової бінаризації. Порогове значення означає, що визначається константа (поріг), яка бінаризує зображення таким чином, що всі пікселі, значення яких лежить під порогом, будуть встановлені як чорні, а кожен піксель над порогом буде встановлений як білий. Тобто, усі пікселі, що на передньому плані, є об'єктами інтересу. Усі інші визначаються як фонові. Існує верхня порогова бінаризація, коли усі пікселі з яскравістю більшою, за поріг, виділяють як важливі дані. Решта являє собою фоном. Іншим способом є нижня порогова бінаризація. В даному способі важливою інформацією усі елементи, що входять нижче за поріг. Також можна вказати два порогових значення. У цьому випадку визначають: важливими елементами є ті, що входять в інтервал, чи

навпаки. Одним із способом вибору є вибір по гістограмі [6, 7]. Результат відтворення операції порогового вимірювання показано на рисунку 1.4.

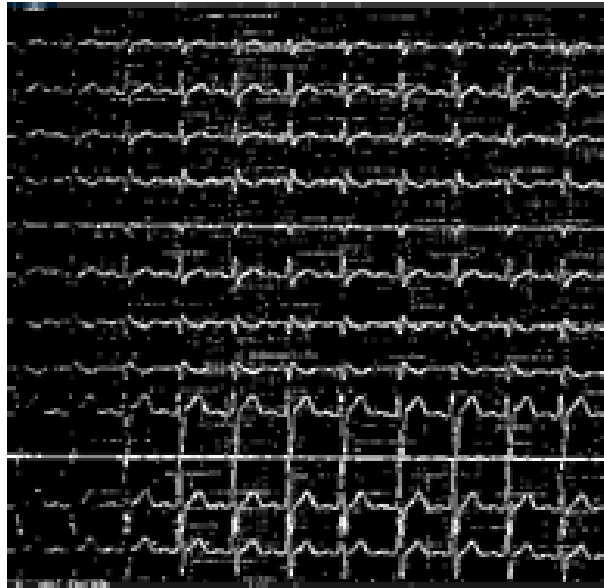


Рисунок 1.4 – Результат операції порогового вимірювання

Оскільки потрібно відтворити кожний окремо сигнал, необхідно знайти спосіб встановити повне розділення сигналів таким чином, щоб інформація про кожен окремий могла зберігатися і отримувати доступ незалежно. Однією з основних проблем під час цієї процедури є виділення перекриваючих сигналів, які траплялись досить часто через високі піки в ЕКГ. Обов'язковим є зменшення втрат якості в процесі розділення. На рисунку 1.5 зображено дані одного сигналу.

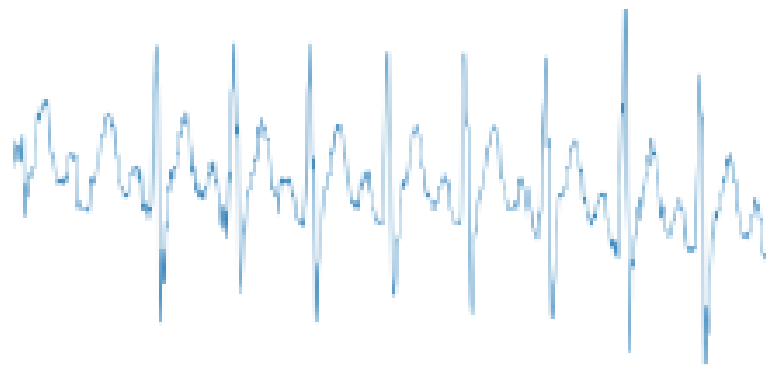


Рисунок 1.5 – Виділення одного сигналу

### 1.3 Методи обробки даних електрокардіограм

Моделі глибокого навчання застосовуються до даних ЕКГ для різних завдань, включаючи виявлення захворювань, анотацію чи локалізацію, постановку сну, біометричну ідентифікацію людини, шум тощо. Глибокі нейронні мережі продемонстрували початковий успіх у серцевій діагностиці за допомогою одномодульних або багатопровідних ЕКГ. Модель глибокого навчання, навчена на великому наборі даних ЕКГ із 91232 записами ЕКГ [1], демонструє кращі показники, ніж кардіологи, для діагностики 12 класів ритму. Для необхідно результату необхідно використовувати точні дані. Для цього потрібно перетворити поточкові сигнали у читабельний формат.

#### 1.3.1 Перетворення у вибірку

Вибірка – метод представлення даних у вигляді числового ряду для подальшого аналізу, класифікації або прогнозування. Для перекладу ЕКГ кожному моменту часу виведення сигналу визначається його числове представлення. Формується числовий ланцюг даних пацієнта. Іншими слова виконується процес дискретизації даних. Сигнал ЕКГ представлений у вигляді неперервної функції, залежної від часу. Іншою залежності використовується шкала, що визначається експертом. Далі на кожному секунду часу виділяється точка на даній кривій та записується її значення по шкалі. Теж саме проводиться для опису аномалій сигналів. Найпростішим способом виявлення аномалій є порівняння ланцюгів між собою. Якщо вибірка, отримана з аномалій, співпадає в даних пацієнта, то вважається, що є відхилення в роботі серця. У випадку, коли подібність є частковою, то вимірюється ступінь схожості. Якщо воно більше за певне порогове значення, то проводиться класифікація та проводиться діагностування на основі отриманих даних. Такий спосіб є найпростішим, що використовується в системах діагностування. Приклад вибірок представлений на рисунку 1.6.

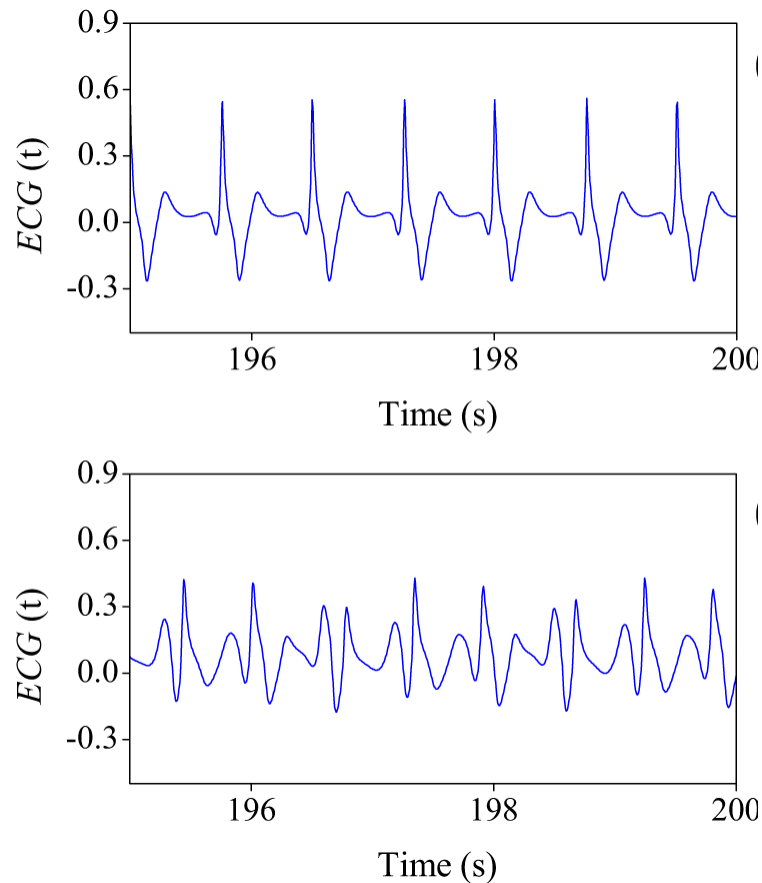


Рисунок 1.6. – Приклади відтворення вибірки ЕКГ

Також використовують аналогічний метод для перетворення сигналу в частоти. Миттєва частота є важливим поняттям в аналізі частоти часу, особливо при аналізі багатокomпонентних сигналів. Дані мають гребінь або концентрацію енергії в площині часових частот навколо миттєвої частоти кожного компонента. Як такі, їх можна використовувати для оцінки зміни частоти ЕКГ (або будь-якого біомедичного) сигналу з часом. Дані подаються у вигляді дискретних точок, що показують рівень сигналу у конкретний час. Зокрема, такий метод використовує для виявлення комплексу QRS та аритмії [8]. Різницею від вище описаного способу є шкала, за якою визначається верхня межа площини. Її значення є не нижче за нуль. Приклад візуального представлення даних ЕКГ у вигляді частот наведений на рисунку 1.7.

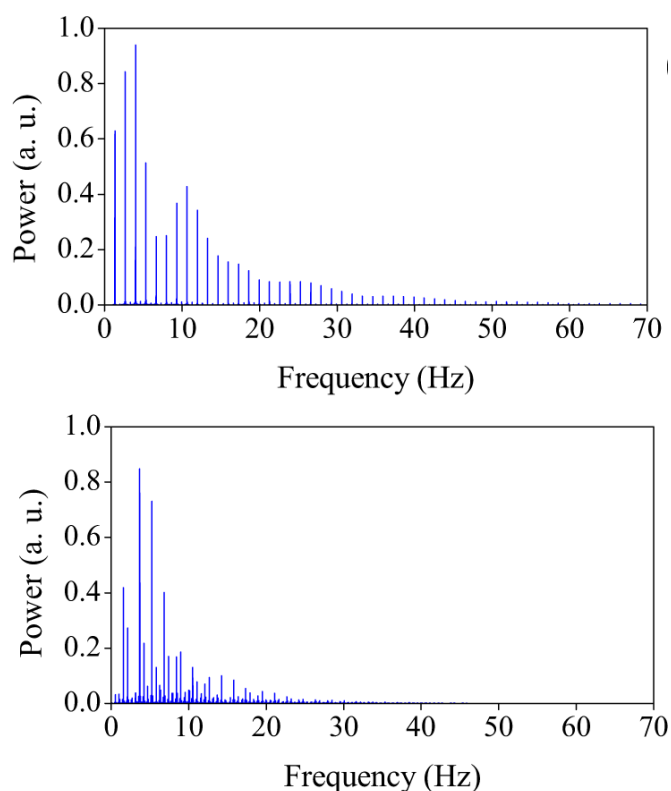


Рисунок 1.7 – Приклади представлення ЕКГ в частотах

Даний метод найчастіше використовується для діагностування хвороб серця і не тільки. Він є простим у реалізації та швидким при розробці системи діагностування. Зокрема використання можна побачити в таких дослідницьких роботах [9,10,11, 12].

Недоліком даного методу є його обмеженість. Використання числових ланцюгів є ефективним способом для аналізу, проте створювати словники чи бази даних для прискорення діагностування є складним етапом. Також існує пряма залежності від процесу обробки зображення ЕКГ. Результат, отриманий від етапу оцифрування даних пацієнта, не підлягає фільтруванню або алгоритмам нормалізації. Тому можливо великий відсоток погрішності підчас виконання етапу безпосередньо класифікації та аналізу. Дані, що будуть подаватись в алгоритми машинного навчання, та є гарантовано правильні та детальні. Щоб запобігти цьому, необхідно використовувати допоміжні алгоритми перетворення та нормалізації.

### 1.3.2 Перетворення Фур'є

Перетворення Фур'є - добре відомий метод, який трансформує сигнал часової області в частотну область для отримання коефіцієнтів частоти. Метод є елементарним перетворенням в цифровій обробці сигналів і має різні застосування в частотному аналізі, обробці сигналів тощо. Це швидкий і спроможний алгоритм для опрацювання дискретного перетворення Фур'є (DFT) і отримує той самий ефект. Визначається формулою, наведеною в рівняння (1.1):

$$X_k = \sum_{n=0}^{N-1} x * e^{-nk2\pi i/n} \quad (1.1)$$

, де  $k$  - ціле число в діапазоні від 0 до  $N-1$ . Сигнали ЕКГ можна стиснути, використовуючи різноманітні методи. Загальний процес складається з наступних етапів:

- отримання зразка ЕКГ або вхідного сигналу;
- стиснення вхідного сигналу шляхом видалення низькочастотних компонентів;
- відновлення вихідного сигналу за допомогою зворотного перетворення Фур'є.

Але недоліком такого методу є те, що він не зміг дати інформацію про точне розташування частотних компонентів у часі [13]. Без гарантії точності наступні етапи систем діагностування будуть виконуватись з погрішністю. Це може привести до неправильного визначення діагнозу. Також дані представлені у вигляді сигналу, що може привести до похибок при аналізі. Даний метод використовується для допоміжної обробки сигналу.

### 1.3.3 Вейвлет-перетворення

Вейвлет-перетворення (Wavelet Transform) з'явилося протягом останніх років як потужний інструмент для аналізу частоти часу та кодування сигналів, сприятливий для опитування складних нестационарних сигналів. Застосування його

до обробки біосигналів було визнано особливо корисним для вивчення цих, часто проблемних сигналів: не більше, ніж ЕКГ. Існує два основних типи WT, відомі як Discrete Wavelet Transform (DWT) і Continuous Wavelet Transform (CWT).

Використовувати даний метод для аналізу не є доцільним, адже дані не є адаптованими для порівняння, аналізу або класифікації. Вони представляються у вигляді сигналу, що може видавати неточності в алгоритмах машинного навчання. Зазвичай його використовують для додаткової обробки даних або перетворення чи фільтрування.

#### *Неперервне вейвлет-перетворення*

Неперервне вейвлет-перетворення (CWT) перетворює безперервний сигнал у сильно надлишковий сигнал двох безперервних змінних: трансляції та масштабу. Отриманий перетворений сигнал легко інтерпретувати у цінний для часового аналізу. Неperервне вейвлет-перетворення безперервної функції,  $x(t)$  відносно дійсного вейвлета,  $\psi(t)$ , описується:

$$W\psi(s, \tau) = \int_{-\infty}^{+\infty} x(t)\psi_{s,\tau}^*(t)dt \quad (1.2)$$

,де

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t-\tau}{s}\right) \quad (1.3)$$

,  $s$  і  $\tau$  називаються параметрами масштабу та перекладу відповідно.  $W\psi(s, \tau)$ , позначає коефіцієнти вейвлет-перетворення, а  $\psi$  - основний материнський вейвлет.

#### *Дискретне вейвлет-перетворення*

Дискретне перетворення вейвлетів (DWT) стало потужною технікою біомедичної обробки сигналів. Його можна записати в тій же формі, що і (1.2), що підкреслює тісний взаємозв'язок між CWT та DWT. Найбільш очевидна відмінність полягає в тому, що DWT використовує значення масштабу і положення на основі

степенів двох. Значення  $s$  та  $\tau$  такі:  $s = 2^j$ ,  $\tau = k * 2^j$  та  $(j, k) \in \mathbb{Z}_2$ , як показано у (1.4) [14].

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t - k * 2^j}{2^j}\right) \quad (1.4)$$

#### 1.3.4 Методи на основі градієнту та морфології

Ще одним із способів є використання градієнту та морфології числової вибірки (Time-Domain Morphology and Gradient). Запропонована методика спрямована на вилучення 11 параметрів, що необхідні, з одного комплексу PQRSST. Повний комплекс PQRSST визначається як послідовність п'яти складених хвиль з деякими відрізками ізоелектричної лінії (100 - 130 мс), включеними на початку (перед Р хвилі) і в кінці (після Т хвилі). Алгоритм TDMG побудований як двоступеневий процес. Перший етап стосується попередньої обробки відібраного комплексу PQRSST, а другий етап є основною функцією вилучення. По суті, 11 випадків, можна згрупувати у дві категорії; моменти часу початку / зсуву та пікові моменти часу. Екземпляри часу початку / зсуву апроксимуються, розглядаючи їх як точки перегину. Через морфологію хвиль ЕКГ (Р, QRS, Т), величина нахилу демонструє різку зміну у випадках початку / зсуву кожної хвилі порівняно з її значенням на відрізках ізоелектричної лінії. Захопивши цю зміну, можна визначити випадки початку та зсуву часу. З іншого боку, пікові моменти часу отримують, розглядаючи ці точки як локальні екстремуми навколо їхнього оточення. Відповідно до стандартної медичної норми, алгоритм TDMG вважає, що пік Q та S завжди є негативними відхиленнями (місцеві мінімуми), тоді як пік R позитивний (локальний максимум). На відміну від цього, Р і Т можуть бути вертикальними (позитивне відхилення), прямолінійними (негативне відхилення) або іноді двофазними (як позитивні, так і негативні сегменти) порівняно з ізоелектричною лінією. Структурна схема алгоритму TDMG зображена на рисунку 1.8.



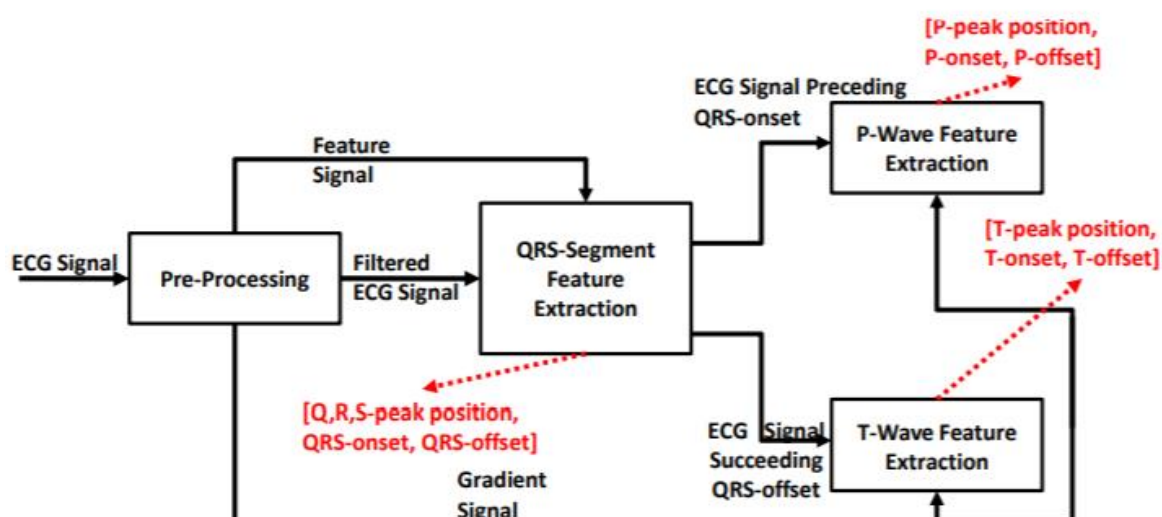


Рисунок 1.8 - Схема алгоритму TDMG

Дві основні цілі, яких намагається досягти алгоритм TDMG, - це точність та повсюдність. Для того, щоб витягнуті ознаки використовувались для медичного обстеження, бажані параметри повинні бути розраховані з максимально можливою точністю. Крім того, алгоритм TDMG розроблений таким чином, щоб бути незалежним від відведення, що використовується для захоплення сигналу ЕКГ, а також мати можливість адекватної роботи на сигналах ЕКГ, відібраних з низькими частотами дискретизації (100-250 Гц) [15].

Недоліком же даного методу є складність в реалізації та швидкості його роботи. Необхідно пройти багато перевірок, щоб виконати перетворення.

#### 1.4 Використання лінгвістичного моделювання

Лінгвістичне моделювання полягає у перетворенні чисельних даних або експериментальних, багатовимірних в лінгвістичну послідовність [16]. Вона являє собою ланцюг із символів або букв, лінгвістичний ланцюг. Лінгвістичним моделюванням є набір методів та алгоритмів для процесу трансформації числових рядів в лінгвістичний ланцюг. За допомогою цього способу можна також проводити процес відновлення даних в первинний вигляд. Основа даного способу моделювання базується на лемі, згідно якої існує ізоморфізм відтворення чисельних даних до лінгвістичних послідовностей.

Лінгвістичною моделлю є представлення лінгвістичного моделювання для конкретної предметної області зі свої властивостями та правилами. Побудова лінгвістичне моделі ґрунтується на трьох основних способах: структурний підхід та математична лінгвістика, інтервальні обчислення та робастні методи, сучасні методи ймовірнісного моделювання. Разом з використанням таких методів можна побудувати та описати правила граматики мова для певної предметної області. Отже, можна створити свою унікальну мову відходячи від конкретного набору чисельних даних. Даний метод підходить для вирішення таких задач, аналіз та прогнозування часових рядів, визначення емоційного стану користувача, задача розпізнавання рухів, рання діагностика хвороб [17].

Щоб отримати готову лінгвістичну модель, необхідно виконати такі етапи роботи [16]:

- попередньо обробити дані, виконати валідацію, видалити шум, профільтрувати;
- розбити масиви даних на менші інтервали, необхідні для перетворення;
- провести лінгвістизацію, тобто, перетворити інтервали в лінгвістичні ланцюги;
- відновити формальну граматику.

### *Структурний підхід*

Ідея структурного підходу полягає у заміні числових значень часового ряду на символи наперед визначеного алфавіту та подальшого виведення правил граматики деякої мови, за якими утворюється даний лінгвістичний ланцюг.

Для початку необхідно розбити дані на такти, щоб отримати дискретне представлення. Далі вони перекладаються в символи визначеного алфавіту. Даний етап виконується шляхом опису правил трансформації елементи в алфавіт. Простим способом є виділення меж інтервалів даних, які будуть відповідати певному символу з алфавіту. Підраховується крок розмежування вертикальної шкали сигналу. Потім будуються інтервали та призначаються кожному певний

символ. В результаті якщо елемент з числового ряду попадає в певний інтервал, то він перетворюється в символ, що йому відноситься за правилами [18].

Після того, як стає відомою граматику, що відповідає нормальному, не аномальному ряду, її співставляти із граматику побудованою на інших значеннях цього ж ряду. Якщо буде помічено значні відхилення, можна сказати, що ці значення є аномальними. Кожен символ алфавіту у лінгвістичному ланцюгу відповідає стану процесу у даний момент часу [19]. У кожному новому рівні ряду система переходить у інший стан із певною ймовірністю. Приклад перекладу показаний на рисунку 1.9, де для спрощення відображення було обрано англійський алфавіт та кожній літері відповідав однаковий проміжок [20, 21].

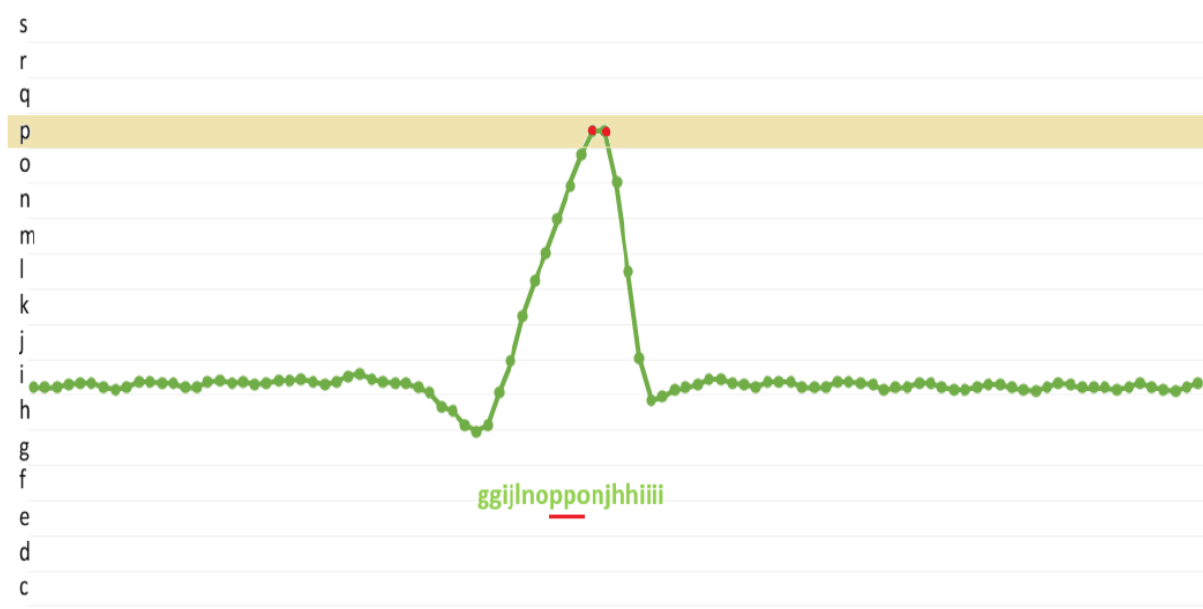


Рисунок 1.9 – Перетворення даних ЕКГ в лінгвістичний ланцюг

Для діагностування хвороб необхідно використати даний принцип на даних пацієнта. Так само необхідно виконати аналогічне перетворення відомостей, що містять аномалії. Наступним етапом є пошук отриманих ланцюгів аномалій в даних пацієнта. Виконати дану задачу можна за допомогою алгоритмів знаходження відстаней між рядками: відстань Левенштейна, відстань Геммінга, Джаро-Вінклера, Дамерау-Левенштейна.

Даний метод є альтернативою представлення та обробку даних чисельного виду. Для подальшого аналізу та прогнозування можна створити словник даних з

аномаліями та без. Це значно прискорить роботу кардіолога у виявленні відхилень від нормального стану серцевого ритму. Адже потрібно буде лише співставляти дані пацієнта зі створеним словником. Такий спосіб детально описаний в дослідженні [4].

Недоліком даного методу є те, що необхідно правильно зазначити граматику та вибрати алфавіт для перекладу сигналу в лінгвістичний ланцюг. Якщо цього не зробити, то є вірогідність або отримати помилкові діагнози, або взагалі не виявляти аномалії. Також отримані результати можуть бути не точні, що приведе до гіршої роботи алгоритмі класифікації.

## 1.5 Постановка завдання

### *Мета*

Основною метою даної роботи є розширення можливостей аналізу даних електрокардіограм за рахунок використання лінгвістичної моделі на основі апарату нечітких множин.

### *Призначення*

Призначення даної роботи є створення програмного забезпечення, що виконує етап попередньої обробки даних ЕКГ за допомогою використання лінгвістичної моделі та апарату нечітких множин, яке має надавати зручний інтерфейс, коректно виконувати усі етапи алгоритму та зберігати результати.

### *Задачі*

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- створити функції зчитування даних та розбиття на такти сигнал ЕКГ;
- удосконалити метод перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання описати функції приналежності та правил нечітких множин;
- створити програмний модуль для реалізації удосконаленого методу;
- створити програмний модуль для візуалізації та збереження результатів обробки;
- дослідити ефективність удосконаленого методу.

Щоб системи діагностування видавали результати з найменшою погрішністю, необхідно використовувати точні та достовірні дані. Для цього обрано метод лінгвістичного моделювання. Проте необхідно правильно задати правила перетворення та обрати алфавіт.

## Висновки до розділу

В даному розділі було описано що являє собою дані електрокардіограми, як вони отримуються. Також розписані складові параметри, по яким проводиться аналіз інформації.

Також описані етапи оцифрування ЕКГ. Основними функціями є видалення шуму, розмірної сутки, рамок, якщо вони є. Далі бінаризація зображення, фільтрування та виділення сигналу для подальшої обробки.

Наведені основні методи роботи відтворення даних для класифікації або пошуку аномалій. Найчастіше всього використовується метод перетворення сигналу в числову вибірку або частоти. Далі такі ланцюги порівнюються та визначається степінь схожості з аномаліями. Недоліком методу є складний етап створення словників таких ланцюгів.

Іншими способами обробки даних є перетворення Фур'є та вейвлет-перетворення. Проте такі методи видають все ще дані у вигляді сигналу, які потім складно аналізувати. Класифікація такої інформації видає неточні результати.

Доцільним є використання лінгвістичного моделювання для даної предметної області. Кожному моменту часу сигналу відповідає певний символ із заданого алфавіту. Формує граматику та своя унікальна мова. Можна створювати словники та виконувати порівняння нових даних із записами. Це значно прискорює аналіз та класифікацію.

Зроблено постановку задачі. Визначено мету, призначення та голові задачі дослідження.

## 2 ЗАСТОСУВАННЯ АПАРАТУ НЕЧІТКИХ МНОЖИН

### 2.1 Визначення поняття нечіткої множини

Процес прийняття рішень в задачах, які пов'язані з управлінням складних динамічних систем, є досить складним або виконується в умовах деякої невизначеності. Це можна охарактеризувати тим, що досить велика частина вхідних даних представляє собою інтервальні або лінгвістичні значення. Їх оцінка є досить суб'єктивною. Така невизначеність може ускладнювати використання двозначної логіки для формулювання задач, пов'язаних з прийняттям рішень під час створення систем підтримки прийняття рішень. Як варіант для вирішення проблеми можна використовувати підхід нечітких множин та нечітких логічних систем, щоб побудувати математичну модель процесів прийняття рішень з урахуванням невизначеності [22].

Математична теорія нечітких множин була запропонована Л. Заде, американським математиком і логіком з українським корінням. Вона дозволяє описувати нечіткі поняття та знання, оперувати цими знаннями та робити нечіткі висновки [23].

Нечіткою підмножиною універсальної множини  $R$  [24, 25] визначається як сукупність множини  $U$  та функції приналежності  $\mu_U(x)$  ( $x \in R$ ), такої, що  $0 \leq \mu_U(x) \leq 1$  ( $\forall x \in R$ ). Вона являє собою набір правил, які визначають чи належить елемент множині, чи ні. Нехай такий набір позначається як  $P$ . Якщо множина  $U$  є скінченною або зліченною, то форма запису нечіткої підмножини має вигляд:

$$U = \{(x, \mu_P(x)) | x \in R\} \quad (2.1)$$

, де  $\mu_P(x)$  – це характеристична функція приналежності елементу  $x$  нечіткій множині  $U$ .

$$\mu_P(x) = \begin{cases} 0, & x \notin U \\ (0, 1), & x \text{ частково} \in U \\ 1, & x \in U \end{cases} \quad (2.2)$$

Характеристична функція належить інтервалу  $[0,1]$ . Вона набуває значення 0, якщо елемент не належить нечіткої множини. Якщо елемент входить в інтервал та зустрічається частіше інших в проміжках, то вважається, що він повністю належить. В такому випадку функція приналежності його дорівнює 1. Усі інші елементи, що входять в нечітку множину, але зустрічаються рідше, будуть мати функцію, що належать проміжку  $(0, 1)$  [26].

Таку «характеристичну функцію», яка залежить від експертної оцінки, називають функцією приналежності, а саму множину для якої цю функцію визначають нечіткою множиною.

Теорія нечітких множин та її застосування полегшує обробку нечітких знань людських експертів у комп'ютеризованій формі. Системи, що базуються на знаннях, такі як експертні системи, а також системи, що базуються на правилах, такі як нечіткі логічні системи, явно використовують базові знання, які, таким чином, безпосередньо помітні та зрозумілі [27]. Впроваджені знання структуровані відповідно до специфікації своєї проблеми, так що легко вибрати конкретні знання, що представляють певну область застосування. Традиційна логіка, яка використовується в експертних системах, присвоює рівно одне з двох можливих значень істини - істинне чи хибне, так чи ні, одне або нуль - кожному твердженню. Багатозначна та нечітка логіка використовують дискретних або неперервних значення відповідно, щоб забезпечити семантичну інтерпретацію тверджень, виразів або формул, які не є ні чітко істинними, ні хибними. Реалізовані функції належності в модулі нечіткості представляють зв'язок між використовуваними лінгвістичними виразами та відповідною областю значень.

## 2.2 Типи функцій приналежності

Функцією належності називають функцію, що визначає ступінь належності елемента до певного інтервалу. Ступінь належності обмежується значенням 0 та 1, де 0 означає, що елемент не входить в множину. Тоді як 1 означає, що елемент входить в множин та зустрічається частіше всього або навіть повністю заповнює

інтервал, що перевіряється. Нечітка множина, що містить елементи із функціями приналежності тільки 1 називають нечіткою одиничкою.

Функції приналежності використовуються на етапах фузифікації та дефузифікації FLS (нечітка логічна система), щоб зіставити нечіткі вхідні значення з нечіткими лінгвістичними термінами і навпаки. Залежно від типу функції приналежності, будуть отримані різні типи нечітких наборів. Заде запропонував ряд функцій членства, які можуть бути класифікуються на дві групи: ті, що складаються з прямих ліній «Лінійні» та «криволінійні» або «нелінійні». Однак нелінійні функції збільшують час обчислення. Тому на практиці більшість програм використовують лінійні відповідні функції.

### 2.2.1 Трикутна функція

Дана функція визначається нижньою межею  $a$ , верхньою межею  $b$  та модальним значенням  $c$ , так що  $a < b < c$ . Існує варіант, коли  $c-b$  межа дорівнює значенню  $b-a$ . У такому випадку використовується поняття симетрична трикутна функція,. Формула даної функції має вигляд:

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & c \leq x \end{cases} \quad (2.3)$$

На рисунку 2.1 зображено графік функції даного типу.

Трикутна функцію відноситься до типу нелінійних функцій приналежності. Вона є простою для задання правил, тому часто зустрічається при формуванні систем з нечіткою логікою. Максимальне значення 1 вона досягає у випадку, коли  $x = b$ . Якщо ж елемент не входить в проміжок  $[a, c]$ , то функція набуває значенню 0.

Також дану функцію можна вважати частковим випадком трапецієвидної функції, визначення якої наводиться в наступному пункті.



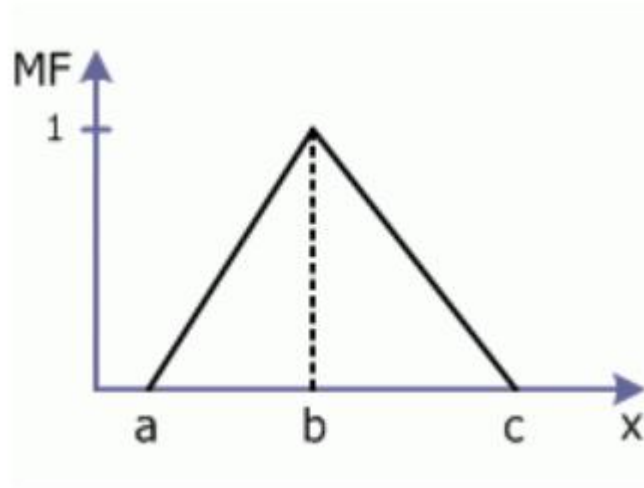


Рисунок 2.1 – Графік трикутної функції

### 2.2.2 Трапецієвидна функція

Трапецієвидна функція визначається нижньою межею  $a$  та верхньою межею  $b$ , а також нижньою та верхньою межами його ядра,  $d$  та  $c$  відповідно. Якщо елемент попадає в інтервал  $[b, c]$ , то функція приймає значення 1, тобто елемент повністю входить в нечітку множину. Формула функції даного типу має такий вигляд:

$$f(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a < x < b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & c < x < d \\ 0, & d \leq x \end{cases} \quad (2.4)$$

Зображення графіка даної функції показано на рисунку 2.2.

Якщо граничні точки  $b = c$ , то в такому випадку функція стає трикутною. Трапецієвидна функція, як і попередня, відноситься до нелінійного виду. Якщо елемент не входить в проміжок  $[a, d]$ , тоді функція набуває значення 0. Завдяки тому, що задається більше правил для визначення елементу множині, дана функція видає більш точні результати для проміжків, що надходять. В даному випадку використовується більше граничних елементів, щоб задати проміжки для опису нечіткої множини.

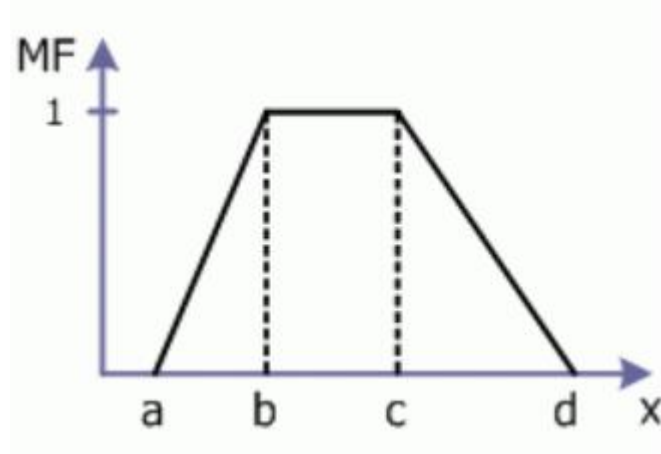


Рисунок 2.2 – Графік трапецієвидної функції

### 2.2.3 S-подібна функція

Дана функція характеризується нижньою межею  $a$ , верхньою межею  $b$  та величиною  $m$  або точкою перегину так, що  $a < m < b$ . Типовим є значення  $m = (a + b) / 2$ . Зростання виконується повільніше, коли відстань  $a - b$  збільшується. Математично S-подібна функція задається такими рівняннями:

$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ 2 * \left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 1 - 2 * \left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 1, & b \leq x \end{cases} \quad (2.5)$$

$$f(x; a, b) = \begin{cases} 0, & x \leq a \\ \frac{1}{2} + \frac{1}{2} * \cos\left(\frac{x-b}{b-a}\pi\right), & a < x \leq b \\ 1, & b < x \end{cases} \quad (2.6)$$

Дана функція класифікується як нелінійна, проте вона не містить точок, що змінюють напрямок зростання, порівнюючи з вище наведеними: трапецієвидна та трикутна.

Графік даної функції має форму, подібно літері S, тому і має таку назву. Її графік зображено на рисунку 2.3.

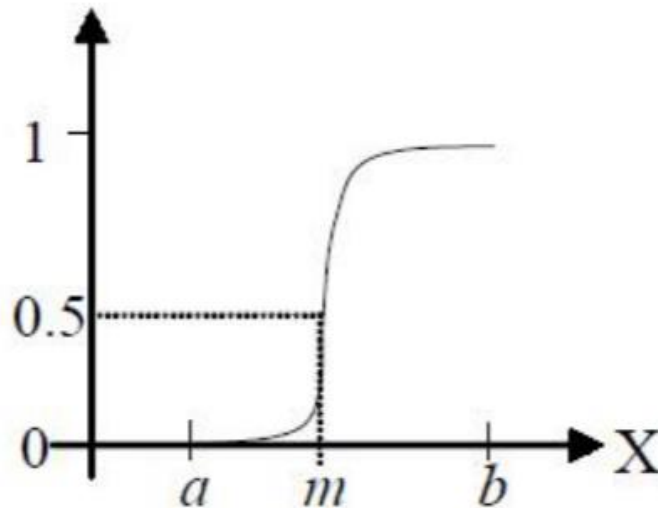


Рисунок 2.3 – Графік S-подібна функції

#### 2.2.4 Z-подібна функція

Остання функція, яка досліджувалась, є Z-подібна функція. Вона визначається нижньою межею  $a$ , верхньою межею  $b$ . Спадання виконується повільніше, коли відстань  $a - b$  збільшується. Дана функція також відноситься до нелінійних функцій приналежності. Математично Z-подібна функція задається такими рівняннями:

$$f(x; a, b) = \begin{cases} 1, & x \leq a \\ 1 - 2 * \left(\frac{x-a}{b-a}\right)^2, & a < x \leq \frac{a+b}{2} \\ 2 * \left(\frac{x-a}{b-a}\right)^2, & \frac{a+b}{2} < x < b \\ 0, & b \leq x \end{cases} \quad (2.7)$$

$$f(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{1}{2} + \frac{1}{2} * \cos\left(\frac{x-b}{b-a}\pi\right), & a < x \leq b \\ 0, & b < x \end{cases} \quad (2.8)$$

Графік даної функції має форму, подібно літері Z і зображено на рисунку 2.4. Основною її відмінністю від S-подібної функції є те, що графік даної спадаю, в той час як попередньої зростає.

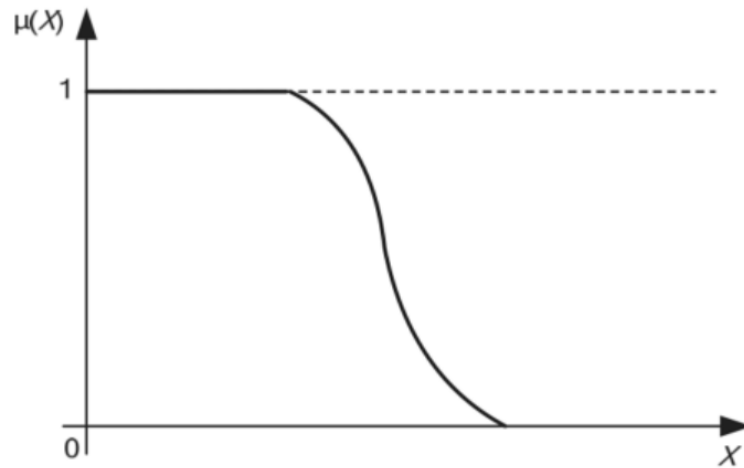


Рисунок 2.4 – Графік функція Z-подібної

### 2.3 Використання функція приналежності на основі частот

Маємо інтервал сигналу  $X(x_1, x_2, \dots, x_m)$ , сукупність нечітких множин  $A(a_1, a_2, \dots, a_n)$ .  $N_a$  – сумарна кількість точок, що задовольняють умови та правила входження в певну чітку множину.

Нехай функція  $v(x)$  показує кількість входження елемента  $x$  в проміжок  $a$ . Тоді функція приналежності на основі частот матиме вигляд:

$$\mu_a(x) = v_a(x) / N_a \quad (2.9)$$

Тому виходить, що чим більше разів елемент попадає у визначений проміжок, тим більше його функція приналежності. Проте вона задовольняє умови належності в проміжок  $[0, 1]$ . Якщо просумувати усі функції приналежності, що є відповідними до певної нечіткої множини  $a$ , то отриманий результат може бути більше за значення 1.

Дану функцію складно представити у вигляді графіку та описати чітке математичне формулювання, оскільки вона напряму залежить від вхідних даних.

Якщо певний елемент має функцію приналежності більше 0 для двох різних нечітких множин, то вибирається та, функція якої є найбільшою.

## 2.4 Система нечіткої логіки

Область нечіткої логіки та теорії нечітких систем базується на уявленні, що деякі відносини вхід-вихід не є “чіткими”. Якщо розглядати процес, коли певна маніпульована зміна вхідного сигналу може призвести до, можливо, трьох різних величин змін у вихідному показнику: низький, середній та високий, то нечітка логіка може забезпечити деяке згладжування, щоб вказати, що результат може бути сумішшю низького та середнього, наприклад [28].

Нечітку логіку можна визначити як нелінійне представлення вхідних даних в залежності від заданих правил формування нечітких множин. Даний метод нагадує міркування людини. Даний підхід подібного до того, як виконує рішення людина. Це включає всі можливі проміжні значення між відповідями «правдиве» і «хибне».

До характеристик систем з використанням нечіткої логіки можна віднести такі, як:

- гнучкість та простота у розширенні функціоналу;
- використання у випадку невизначених або наближених міркувань;
- імітування логіки людської думки;
- побудова нелінійних функцій довільної складності.

Апарат нечітких множин складається з 4 основних компонентів.

Першим етапом є визначення бази правил. Вона містить набір правил та умови IF-THEN, надані експертами для управління системою прийняття рішень на основі поданої інформації. Останні розробки в теорії нечітких теорій пропонують кілька ефективних методів для проектування та налаштування нечітких контролерів. Більшість із цих розробок зменшують кількість нечітких правил.

Другим компонентом є етап фазифікації. На даному кроку виконується конвертування вхідних даних. На вхід подаються чітко визначенні дані, наприклад числовий ряд, частоти, заміри датчиків, тощо. Перед виконанням трансформації необхідним є також процес обробки даних, що надходять. Виконуються такі кроки, перетворення сигналу, видалення шуму або непотрібних даних, нормалізація по мірі необхідності.

Третім компонентом є механізм логічного перетворення, який представляє собою алгоритм конвертування вхідних даних у вихідні. Він визначає ступінь узгодження поточного нечіткого введення щодо кожного правила та вирішує, які правила потрібно запускати відповідно до вхідних даних. Далі обрані правила поєднуються для формування контрольних дій. Якщо надходжений елемент виконує усі умови правил, що описані для певної нечіткої множини, то він трансформується у вихідний вигляд. Функція приналежності такого елементу має набувати значення 1 або максимальне, відмінне від 0. Вихідний результат залежить від предметної області та проблеми, що вирішує система нечіткої логіки. В даному дослідженні це є перетворення в лінгвістичний вигляд.

Останнім компонентом є дефазифікація. На даному етапі сумують вихідний результат та виводять його в потрібному форматі. Він використовується для перетворення нечітких множин, отриманих за допомогою механізму логічного перетворення, у чітке значення. Доступно декілька методів дефазифікації, і найкращий із них використовується із спеціальною експертною системою для зменшення помилки [29, 30].

На рисунку 2.5 зображено архітектуру компонентів системи, що використовує нечітку логіку.



Рисунок 2.5 – Архітектура системи на основі нечіткої логіки

Перевагами у використанні системи даного типу можна навести такі [25]:

- структура системи є простою та зрозумілою;

- гнучкість та простота в інтеграції в інші системи та машинне навчання;
- описує невизначеність в проблемі;
- легкість в модифікації для поліпшення або зміни продуктивності системи;
- більш ефективніше вирішення складних питань або проблем.

До недоліків можна віднести такі:

- результати не можуть бути загальновизнаними, оскільки вони визначаються на основі припущень;
- необхідно детальне тестування апарату нечіткої множини експертами;
- встановлення точних правил та функції приналежності є досить складним завданням;
- іноді нечітку логіку плутають з теорією ймовірності.

Системи з нечіткою логікою є досить розповсюдженими у комерційному використанні. Наприклад компанія Mitsubishi Chemical використовує даний апарат у мікрохвильових печах для встановлення потужності випромінювання та спосіб приготування. Також компанія Canon використовує в принтерах для визначення напруги барабану на основі щільності зображення, вологості та температури. Компанія Nissan використовує в системах антиблокування гальм для управління ними в небезпечних випадках, що залежить від швидкості автомобіля та прискорення.

## 2.5 Удосконалений метод на основі апарату нечітких множин

Метод перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання функції приналежності та правил нечітких множин складається з таких етапів:

- 1) розбити такт на інтервали з однаковою кількістю елементів;
- 2) кожному отриманому інтервалу задати символ;
- 3) підрахувати функції приналежності кожного елемента в певний інтервал;

4) перетворити елемент у той символ, функція приналежності якого є найбільшою.

В результаті застосування методу отримуємо наступні дані:

- лінгвістичний ланцюжок;
- список інтервалів;
- кількість елементів, що входить в інтервал;
- суму функцій приналежності всіх елементів, що входять в інтервал.

Схема алгоритму наведена у додатку А. Ефективність методу наведено у підрозділі 3.5.

Метод відкриває такі можливості аналізу даних на основі апарату нечітких множин:

- застосування аналізу на основі порівняння лінгвістичних ланцюгів даних пацієнта та аномальних;
- класифікація даних за допомогою нейронних мереж;
- використання словників для прискорення процесу виявлення аномалій.

## Висновки до розділу

У даному розділі було визначено поняття нечіткої множини та характеристичної функції. Представлено математичне забезпечення функції призначення та правил задання нечітких множин. Визначено методи застосування апарату нечітких множин в системах.

Детально розглянуто типи функцій приналежності, а саме: трикутна, трапецієвидна, S-подібна та Z-подібна, їх математичне формулювання, показано їх графіки. Усі дані типи належать до функцій приналежності нелінійної класифікації. Вони досить часто використовуються при розробці систем з нечіткою логікою.

Описано формування функції приналежності на основі підрахунку кількості входження кожного елемента в певну нечітку множину. Даний спосіб використовується для реалізації програмного забезпечення попередньої обробки електрокардіограм.



Визначено поняття системи, що використовує нечітку логіку. Сформульовано її характеристики. Такі системи складаються з 4 компонент: база правил, фазифікація, механізм логічного перетворення, дефазифікація. Наведено та описано основні етапи для створення даних систем. Визначено переваги та недоліки у використанні апарату нечітких множин в системах прийняття рішень.

Наведено етапи, з яких складається удосконалений метод лінгвістичного моделювання на основі апарату нечітких множин. Визначено результативні дані та можливості аналізу даних, що відкриває даний метод.

### **3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РЕЗУЛЬТАТИ**

#### **3.1 Вимоги до програмного забезпечення**

Програмне забезпечення попередньої обробки даних ЕКГ має призначення першого етапу в роботі експертно-діагностичної системи. Використання даної програми як самостійної не є доцільним, так як результати, що отримуються, можуть бути не зрозумілими для лікарів або медичних експертів. Тому важливим значенням є здатність інтегрування програми в інші системи діагностики. Вона має виконувати усі поставлені завдання, що описані в попередніх розділах.

Необхідно точно сформулювати вимоги, щоб мати змогу розробити програму, яка буде вирішувати дану проблему. Для точного опису було виділено функціональні та нефункціональні вимоги.

До функціональних вимог даного ПЗ відносяться:

- виконання зчитування даних із вхідних файлів формату CSV;
- правильне виділення R-піків та одного повного циклу серцебиття;
- виконання алгоритму перетворення даних в лінгвістичних ланцюг;
- можливість збереження правил формування нечітких множин у файл формату JSON;
- збереження результатів перетворення у файл формату RTF;
- візуалізація результатів;
- перетворення лінгвістичного ланцюга в числовий ряд для оцінки точності роботи програми.

Дані функціональні вимоги можна поєднати в модулі, такі як: зчитування даних та розбиття на такти, виконання перетворення, збереження результатів, виконання оберненого перетворення для оцінки ефективності.

До нефункціональних вимог можна віднести такі:

- можливість запуску програми в операційній системі Windows;
- представлення зручно та інтуїтивного інтерфейсу для взаємодії з програмою.

Важливим етапом при розробці даного програмного забезпечення є тестування його та оцінка ефективності алгоритму. Метою даного дослідження є вдосконалення вже існуючого методу, тому щоб оцінити роботу запропонованого, слід розробити модуль порівняння результату оберненого процесу даного методу та вже існуючого.

### 3.2 Засоби розробки

Для розробки програми було вирішено використовувати мову програмування C# для написання функціональних модулів. C# - це сучасна, об'єктно-орієнтована мова програмування загального призначення, розроблена корпорацією Майкрософт та схвалена Європейською асоціацією виробників комп'ютерів (ECMA) та Міжнародною організацією стандартів (ISO). C# призначений для Common Language Infrastructure (CLI), яка складається з виконуваного коду та середовища виконання, що дозволяє використовувати різні мови високого рівня на різних комп'ютерних платформах та архітектурах. Дана мова використовує безпечну систему типізації для платформи .NET [31]. Вибір даної мови для розробки ПЗ було зроблено на основу таких факторів:

- використання об'єктно-орієнтованого способу програмування;
- автоматичний збір сміття та очищення пам'яті;
- кросплатформенність;
- безкоштовне використання компілятора та середовища виконання;
- доступність багатьох ресурсів для пошуку інформації;
- можливість використання великої кількості бібліотек для реалізації функціоналу, зокрема для роботи з CSV та JSON файлами;
- наявність строгої типізації.

Для розробки інтерфейсу роботи користувача з програмою було вирішено використовувати WindowsForm. WindowsForm – бібліотека платформи .NET саме для програмування інтерфейсу. Функціонал для роботи елементів форми можна розробити, використовуючи мову програмування C#, що значно підходить для

роботи. Програму можна легко та зручно розгорнути на платформі Windows, що також підходить для використання в даній розробці [32].

Однією з переваг є робота з деякими елементами управління, такими як RichTextBox, набагато простіша. Менше роботи потрібно зробити для простих речей, таких як встановлення розширеного тексту елемента керування.

Також дана бібліотека пропонує меншу кількість налаштувань. Хоча це модно віднести до недоліків, проте не для даної програми. Оскільки в результаті програмне забезпечення використовується як частина системи, не має сенсу зациклюватись на деталях інтерфейсу. Тобто, не потрібно витрачати багато часу на розробку дизайну.

Останнім фактором для вибору WindowsFrom є те, що результативна програма займає менший обсяг пам'яті на комп'ютері. Порівняно з нею такі інструменти, як WPF, займає більше оперативної пам'яті. Це також є актуальним, оскільки багато пам'яті будуть займати саме вхідні та вихідні дані [33].

Розробка виконувалась у інтегрованому середовищі розробки Rider IDEA. Воно включає до свого складу текстовий редактор, компілятор, систему збірки, а також систему контролю версій. Також зручно використовувати аналізатор коду при написанні програми, щоб менше витрачати часу на виправлення помилок та тестування програми з точки зору запуску та компіляції.

### 3.3 Архітектура програмного забезпечення

Програмне забезпечення створюється з ціллю обробити дані перед використанням їх вже в алгоритмах класифікації. Воно має виконувати поставлені задачі. Тому потрібно розробити відповідну архітектуру, щоб досягти мети роботи.

Система програмного забезпечення складається з 5 основних модулів, кожен з яких відповідають своїм задачам:

- модуль зчитування даних;
- модуль розбиття на такти;
- модуль перетворення в лінгвістичний ланцюг;
- модуль збереження результатів;

- модуль інтерфейсу.

Діаграма компонентів наведена в Додатку А, яка ілюструє зв'язки між вказаними компонентами, а також зв'язок із користувачем.

Модуль «Read CSV File» містить необхідний функціонал для зчитування вхідних даних із файлу формату CSV. Для цього використовується безкоштовна та доступна бібліотека `LumenWorks.Framework`. Вона містить усі можливі функції для маніпулювання файлами формату CSV. Для роботи модуля необхідно вказати ім'я вхідного файлу [34].

Модуль «Segmentation» виконує функцію розбиття числового ряду на такти. В даному компоненті реалізується алгоритм розбиття сигналу ЕКГ на менші інтервали. Здійснюється виділення R-піків, тобто максимальних значень, а також визначається сусідні від нього значення. На вхід подається великий масив числових даних, отриманих з попереднього модуля.

Наступним модулем є модуль «Convert to Chain». Він є найбільший за функціональністю та містить найголовніші функції з точки зору роботи даного дослідження. В даному модулі виконується створення апарату нечітких множин та лінгвістичної моделі. На вхід подається інтервал, отриманий з попереднього модуля. Далі створюється база правил, тобто розподіл та задання символу із заданого алфавіту множини елементів. Після цього підраховується функція приналежності кожного елемента конкретній множині. Для цього використовується метод на основі частоти входження, який описаний в попередньому розділі. Якщо результат функції більше за значення 0, то даний елемент трансформується у відповідний символ. В результаті кожний елемент вхідного символного ряду перетворюється в певний лінгвістичний символ, в залежності в його функції приналежності. На виході створюється лінгвістичний ланцюг.

Модуль «Write File» містить функція для запису результатів. Необхідно вказати файл, в який користувач хоче зберегти результати. В нього будуть записані усі лінгвістичні ланцюги, які були отримані із вхідного файлу. Також є можливість записати інтервали, які були отримані із апарату нечітких множин. Ці

результати записуються в файл формату JSON разом з співставленням кожного символу нечіткій множині та її суму всіх функція приналежності [35]. В подальшому ці результати, а також лінгвістичні ланцюги, можна використовувати в алгоритмах класифікації або для створення баз даних лінгвістичних ланцюгів із даними за аномаліями.

Останнім модулем є інтерфейс користувача «GUI». Він складається з функцій, що пов'язують роботу усіх інших компонентів з елементами форми. Запуск програми виконується за допомогою файлу Program.cs. Він містить стартовий код. Файл Form.cs містить інформацію про усі компоненти форми та функції роботи елементів при змінні стану форми. Він є головним компонентом для взаємодії клієнта з програмою. Користувач шляхом маніпулювання графічними елементами може задавати вхідні дані, зберігати результати у вказані файли та побачити візуально вихідний лінгвістичний ланцюг [32, 33].

Архітектура даного програмного забезпечення є зрозумілою та має усі необхідні компоненти для реалізації роботи удосконаленого методу обробки даних електрокардіограм. З використанням даної архітектури має бути реалізоване комп'ютерне програмне забезпечення, що надає можливість виконувати усі завдання для досягнення мети за допомогою зручного інтерфейсу та програмного комплексу удосконаленого методу.

### 3.4 Опис експерименту

#### 3.4.1 Опис вхідних даних

Експериментальні дослідження здійснювались із застосуванням описаного вище програмного забезпечення. В якості експериментальних даних використовувався набір сигналів ЕКГ, що мають вигляд числового ряду. На рисунку 3.1 показано приклад вхідних даних, що зчитувались з файлу формату csv. Вони представлені у вигляді ряду дійсних чисел. При необхідності додається колонка з мітками аномалій. Їх візуальне представлення показано на рисунку 3.2.

	A
1	-0.025024
2	-0.028381
3	-0.031433
4	-0.033875
5	-0.036621
6	-0.038452
7	-0.040588
8	-0.042114
9	-0.043335
10	-0.045166
11	-0.046997
12	-0.048218
13	-0.048523
14	-0.048523
15	-0.048523
16	-0.048218

Рисунок 3.1 – Приклад вхідних даних у файлі формату CSV

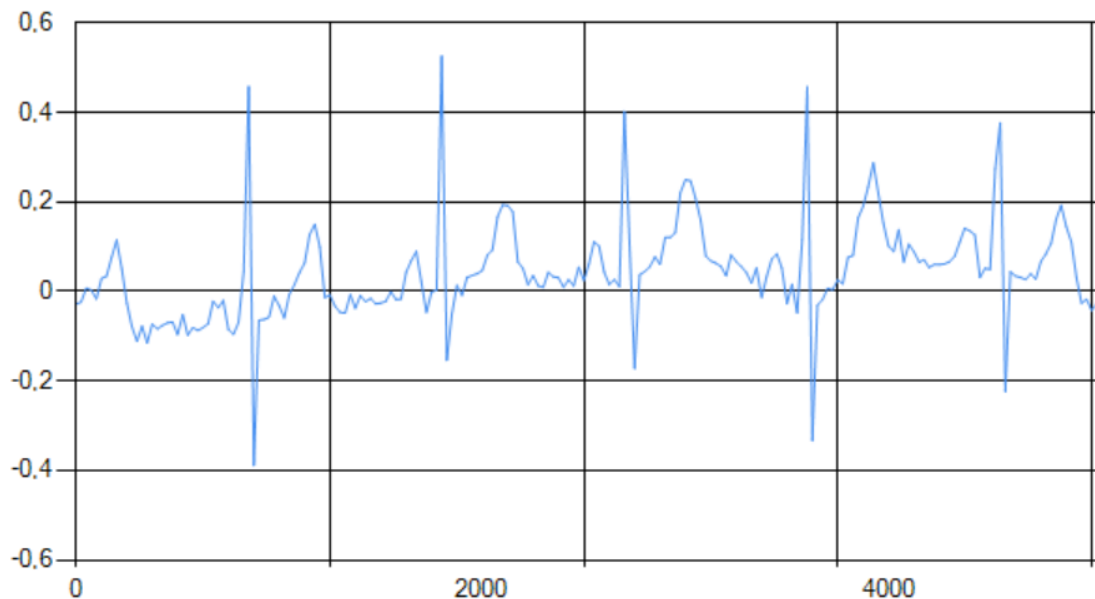


Рисунок 3.2 – Візуальне представлення вхідних даних

Далі проводиться алгоритм сегментації або розбиття на цикли серцевого ритму. Визначаються R піки та сусідні від них хвилі. В інтерфейсі користувача результат даного модуля не виводиться, проте візуально результат можна представити як на рисунку 3.3.

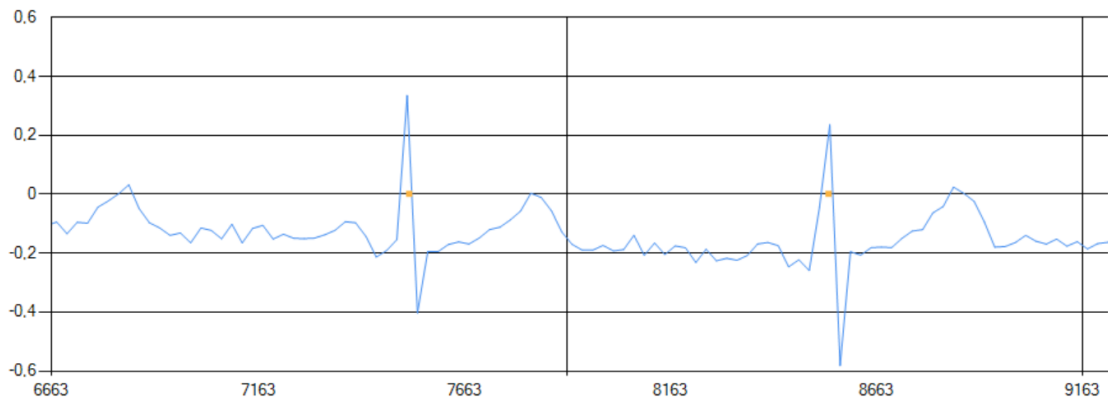


Рисунок 3.3 – Розділення на такти сигнал ЕКГ

### 3.4.2 Опис вихідних даних

В результаті виконання роботи програми зберігається результат створення апарату нечітких множин та файл із перетвореними лінгвістичними ланцюгами.

Дані про правила формування нечітких множин та їх відношення до символів алфавіту зберігаються у файлі формату JSON [35]. Для того, щоб провести серіалізації даних, перш за все інформація зберігається в проміжну сутність, що являє собою клас `FuzzySets`. Файл містить інформацію про номер сегменту інтервалу, які були отримані в результаті проведення сегментації вхідних даних. Далі міститься детальна інформація про кожний сегмент. Вказується початкове та кінцеве значення нечіткої множини, її символ, сумарну функцію приналежності всіх елементів та кількість елементів, що входить в множину. Його структуру наведено в таблиці 3.4.1. Виходячи з цих даних можна виконати обернене перетворення кожного сегменту в дані початкового вигляду, тобто, числовий ряд.

Таблиця 3.1 – Структура збереження даних нечітких множин

Назва властивості	Тип	Опис
Segment_id	Int	Містить інформацію про номер сегменту розподіленого сигналу, який перетворюється в лінгвістичний ланцюг



Продовження таблиці 3.1

Назва властивості	Тип	Опис
Start_point	Double	Вказує первинне значення інтервалу нечіткої множини. Елементи, що дорівнюють цьому значенню або більше за нього, виконують першу умову входження в множину
End_point	Double	Вказує останнє значення інтервалу нечіткої множини. Елементи, що дорівнюють цьому значенню або менші за нього, виконують другу умову входження в множину
Symbols	Char	Вказує на символ, якому характеризується нечітка множина
Member_function	Double	Показує суму функцій приналежності всіх елементів, що входить в множину. Значення може перевищувати 1
Elements_count	Int	Показує кількість елементів, що входить в нечітку множину

При перетворенні елемента в символ виконується аналіз виконання усіх вимог входження в нечітку множину. Задається інтервал певної множини, тому що трансформувати певне значення ЕКГ в правильний символ, необхідно, щоб він входить в проміжок. Іншими словами елемент  $X$  має задовольняти таку вимогу

$\text{Start\_point} \leq X \leq \text{End\_point}$ . Якщо воно так, то розраховується його функція приналежності та виконується переклад в символ.

Експеримент використовувався з алфавітом кількістю 52 символи. Алфавіт мав значення усі латинських букв верхнього та нижнього регістрів. З використанням даних про нечіткі множини виконувалась трансформації на основі обраної множини символів. Результат зображено на рисунку 3.4. Показано візуальне представлення сигналу, горизонтальний розподіл на нечіткі множини, символи, що відповідають множинам та виділення значень R-піків.

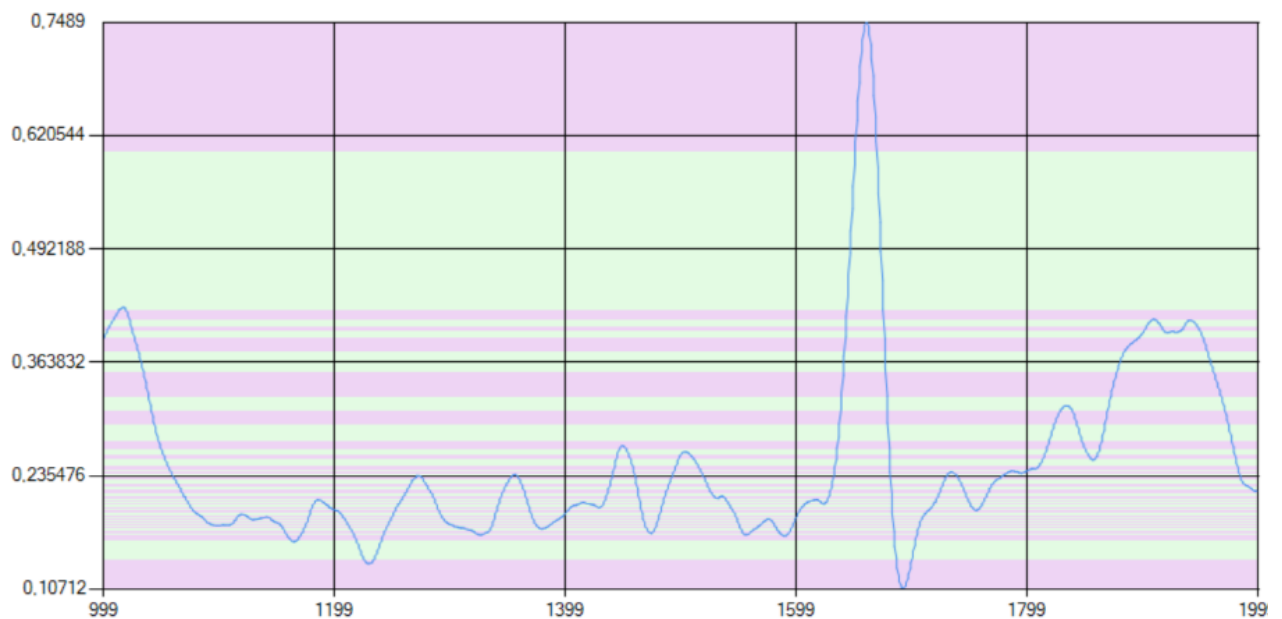


Рисунок 3.4 – Візуальне представлення результату перетворення в ланцюг символів

Також в окремому полі виводиться результат перетворення числового інтервалу в сам лінгвістичний ланцюг. Приклад таких даних показано на рисунку 3.5.

Лінгвістичний ланцюг  
romkigfecbaaaaaabdcdfijmqsux&DFIKMNOPQSTUVWXYXXYYXXWVUTUUUVWXXYZZZZYYXWWUTSQNMJFDAywwusrq  
ppqrrtuvx

Рисунок 3.5 – Вивід лінгвістичного ланцюга

Останнім етапом є збереження результату в файл формату rtf. Приклад такого збереження наведений на рисунку 3.6.

Segment 1

```
romkigfecbaaaaabcdfijmnsuxADFIKMNOPQSTUVWXYXXVWVUTUUUVWXYZZZYXWWUTSQN
MJFDAywusrqppqrrtuvx
```

Segment 2

```
romkigfecbaaaaabcdfijmnsuxADFIKMNOPQSTUVWXYXXVWVUTUUUVWXYZZZYXWWUTSQN
MJFDAywusrqppqrrtuvx|
```

Segment 3

```
romkigfecbaaaaabcdfijmnsuxADFIKMNOPQSTUVWXYXXVWVUTUUUVWXYZZZYXWWUTSQN
MJFDAywusrqppqrrtuvx
```

Segment 4

```
romkigfecbaaaaabcdfijmnsuxADFIKMNOPQSTUVWXYXXVWVUTUUUVWXYZZZYXWWUTSQN
MJFDAywusrqppqrrtuvx
```

Рисунок 3.6 – Результат збереження в файл

### 3.5 Дослідження ефективності

Для оцінки ефективності було реалізовано окремий модуль, який виконував зворотній процес перетворення. На вхід подавались дані у вигляді лінгвістичного ланцюга, на виході було отримано їх числовий ряд. Далі виконується порівняння отриманого ланцюга з початковими.

Реалізується пошуку відхилення за допомогою методу середньо квадратичного відхилення. Даний метод є досить поширеним для значення відхилення числових даних. Перевагою у використанні є те, що, якщо різниця результативних та початкових даних буде від'ємною, то вона підноситься до квадрату, значно впливає на результат порівняння.

Також був реалізований вже наявний метод попередньої обробки даних ЕКГ за допомогою лінгвістичного моделювання, який описаний в роботі [3]. Разом з ним створений метод оберненого перетворення та визначення середньоквадратичної похибки даних.

Алгоритм даного модуля представлений на рисунку 3.7.



Рисунок 3.7 – Алгоритм процесу дослідження ефективності

Результат алгоритмів визначень похибок вказаним методом представлений в таблиці 3.2.

Таблиця 3.2. – Результат дослідження ефективності

Номер сегменту сигналу ЕКГ	Похибка методу зі структурним підходом	Похибка методу з апаратом нечітких множин	Різниця
1	8,357%	6,161%	2,196%
2	7,836%	7,472%	0,364%

Продовження таблиці 3.2

Номер сегменту сигналу ЕКГ	Похибка методу зі структурним підходом	Похибка методу з апаратом нечітких множин	Різниця
5	3,044%	1,937%	1,107%
10	3,852%	1,113%	2,739%

Аналізуючи результати, що наведені в таблиці 3.2, метод лінгвістичного моделювання з апаратом нечітких множин має менший відсоток похибки при оберненому перетворенні, порівнюючи зі структурним підходом. Тому запропонований метод можна вважати, за дає більш точну інформацію в лінгвістичних ланцюгах. Середньоквадратична похибка зменшилась на 1,6%.

### Висновки до розділу

Визначено функціональні вимоги та нефункціональні вимоги до розробки програмного забезпечення. До нефункціональних вимог винесено наявність зручного інтерфейсу та можливість розгортання програми на операційній системі Windows. До функціональних вимог усі функції, що реалізують задачі, які описані в попередніх розділах. Визначено програмні засоби, що будуть використовуватись для розробки, а також наведено переваги їх використання.

Детально розписано архітектуру програмних компонентів. Виділено 5 головних модулів: зчитування даних, розбиття на такти, перетворення в лінгвістичний ланцюг, збереження результатів та останній модуль представлення інтерфейсу. Описано яку функцію виконує кожний компонент та що входить в нього.

Було створено функцію зчитування даних та модуль розбиття на інтервали сигнал ЕКГ. Розроблений програмний модуль для реалізації методу перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання описати функції

приналежності та правил нечітких множин. Створений програмний модуль для візуалізації та збереження результатів обробки.

Описано та наведено приклади вхідних даних та їх візуальне представлення. Також описано в якому вигляді подаються та зберігаються результати. Розписано структуру файлу JSON, в якому зберігається інформація про апарат нечітких множин. Наведено приклад результату візуалізації вихідних даних та приклад файлу, в який записується інформація.

Описано етап дослідження ефективності. Наведено алгоритм процесу дослідження ефективності. Етап дослідження ефективності проводився методом оберненого перетворення. Реалізований метод, що здійснював обернене перетворення лінгвістичного ланцюга в числовий ряд. Потім виконувався процес порівняння результуючих даних із вхідними за допомогою методу визначення середньоквадратичної похибки. Ці ж самі кроки виконувались для алгоритму обробки ЕКГ з використанням лінгвістичного моделювання на основі структурного методу. Показані результати порівняння.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї

За ідею проекту було обрано створення програмного забезпечення, що буде виконувати попередню обробку даних електрокардіограм та зберігати результати лінгвістичних ланцюгів для подальшого використання у класифікації та виявленні аномалій.

Цільовою аудиторією є державні та приватні компанії, що створюють апарати класифікації та діагностування хвороб, пов’язаних з роботою серця. А також ІТ компанії, що розробляють експертно-діагностичні системи для подальшого впровадження в медичні заклади.

В таблиці 4.1 наведені зміст ідеї, напрямки застосування та основні переваги, що може отримати клієнт.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмне забезпечення для попередньої обробки ЕКГ	Підсистема в роботі апарату діагностування серцево-судинних хвороб	Пришвидшення процесу встановлення діагнозу лікарями
		Автоматизація процесу діагностування
	Первинний етап виявлення аномалій даних в експертно-діагностичних системах та програмах	Прискорення роботи виявлення аномалій в даних ЕКГ

Після того, як був проведений детальний пошук застосування, що виконують попередню обробку ЕКГ, було виявлено програмне забезпечення Cardiolund ECG Parser. У таблиці 4.2 представлено визначення сильних, слабких та нейтральних характеристик ідеї проекту в порівнянні зі схожим продуктом.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів		W (слаб ка сторо на)	N (нейт раль на сторо на)	S (силь на сторо на)
		Проект	Cardiolund ECG Parser			
1	Сегментація сигналів серцебиття	Наявна	Наявна		+	
2	Формат даних для подальшого використання	Лінгвістичний ланцюг	Числова послідовність			+
3	Інтеграція з іншими програмними забезпеченнями	Наявна	Наявна		+	
4	Збереження результатів перетворення	Наявне	Відсутнє			+



Завдяки аналізу слабких, сильних, нейтральних характеристик та характеристик ідеї потенційного товару та його аналогу можна знайти конкурентоспроможність даної ідеї. Таким чином можна виділити, що даний проект має, порівняно з конкурентом, такі переваги: використання лінгвістичного методу обробки інформації та отримання в результаті файл із збереження даних для подальшого аналізу ЕКГ.

## 4.2 Технологічний аудит ідеї проекту

В даному підрозділі проводиться аудит технології, що показує технології для реалізації ідеї проекту, також визначає їх наявність і доступність. В таблиці 4.3 зазначені які технології необхідні для реалізації ідеї та чи доступні вони для розробки ПЗ.

Таблиця 4.3 - Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Розробка програмного забезпечення	C#	Є наявності у	Технологія доступна
2	Зчитування даних	LumenWorks.Framework	Є наявності у	Технологія доступна
3	Візуалізація	WindowForm	Є наявності у	Технологія доступна
4	Запис даних у файл JSON	Newtonsoft.Json	Є наявності у	Технологія доступна
Обрана технологія реалізації ідеї проекту: в якості середовища розробки було обрано мову програмування C#, для візуалізації було обрано технологію WindowsForm				

Проаналізувавши дані в таблиці можна зробити висновок, що для розробки ідеї не потрібно створювати нові технології. Проект може бути реалізований за допомогою наявних та доступних інструментів.

### 4.3 Аналіз ринкових можливостей запуску стартап-проекту

Необхідно проаналізувати ринкові можливості перед тим, як вивести продукт на ринок. Результат аналізу попередньої характеристики потенційного ринку стартап-проекту наведено в таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1
2	Загальний обсяг продаж, грн/ум. од	Невідома
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Збільшення можливостей серверної частини для покращення роботи в великими обсягами даних та їх зберігання
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги до інтеграції в системи медичних закладів
6	Середня норма рентабельності в галузі (або по ринку), %	Невідомо

За результатами таблиці попереднього аналізу потенційного ринку можна зазначити, сфера аналізу даних ЕКГ є принадливий, а ринок є звабливий для запуску. Хоча наявні деякі обмеження, тому потрібно враховувати специфічні

вимоги до стандартизації та специфікації програмного забезпечення, які пов'язані в медичними послугами.

Важливою частиною дослідження ринкових можливостей є визначення цільової аудиторії та потенційних клієнтів, які будуть використовувати продукт, та їх вимоги до споживання програми (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

<b>Потреба, яку формує ринок</b>	<b>Цільова аудиторія (цільові сегменти ринку)</b>	<b>Відмінності у поведінці різних потенційних цільових груп клієнтів</b>	<b>Вимоги споживачів до системи</b>
Попередня обробка даних сигналів ЕКГ	Приватні та державні компанії, що створюють апарати для центрів кардіології	Технологія може використовуватись для пришвидшення діагностування хвороб, пов'язаних з роботою серця.	Висока швидкість обробки вхідних даних, можливість обробки великої кількості даних, безпомилкове видання результату.
	ІТ компанії, що розробляють експертно-діагностичні системи	Технологія може використовуватись як перший етап в алгоритмах класифікації та розпізнавання аномалій даних в ЕКГ.	Висока швидкість обробки вхідних даних, можливість обробки великої кількості даних, безпомилкове видання результату, документація, розширювальне API.

Аналіз можливостей входження на арену ринку також включає визначення факторів, що можуть перешкоджати надходженню продукту та можливості його впровадження. Проведемо розбір ймовірних ризиків та можливостей стартап-

проекту. Фактори загроз розглянуті в таблиці 4.6. Можливості розглянуті в таблиці 4.7.

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Відмова у співпраці	Діагностичні центри не будуть зацікавлені у використанні продукту	Отримання зворотної оцінки щодо використання продукту, введення змін, запропонування компромісів.
2	Кваліфікація кадрів	Недостатній рівень знань та кваліфікації команди розробки продукту	Пошук нових розробників, долучення експертів для передачі знань.
3	Кошти на розробку продукту	Нестача грошей на розробку програми	Аналіз та оптимізація грошових витрат, залучення нових інвестицій, мотивація роботи на перспективу.
4	Конкуренти	Впровадження схожого продукту конкурентами	Розширення та вдосконалення нового функціоналу, акцентування на сильних сторонах даного продукту, зміна маркетингової стратегії
5	Швидкість роботи	Програма витрачає багато часу на обробку даних	Заміна повільних компонентів система, придбання потужніших серверів, вдосконалення алгоритму.

Продовження таблиці 4.6

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
6	Недостатньо пам'яті	Не вистачить пам'яті для зберігання даних ЕКГ та результатів	Використання хмарних технологій для зберігання даних
7	Недовіра лікарів у точності результатів роботи програми	Лікарі віддають перевагу роботи з паперовими результатами та аналізу даних власноруч	Проведення маркетингової компанії для пояснення роботи програми, а також розвіяння всіх сумнівів щодо неточності результатів.

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Продаж компаніям, що розробляють системи діагностування	Грошове отримання за використання технологій у подальшій розробці систем аналізу	Збільшення функціоналу, розширення компанії
2	Зацікавленість держави у покращенні якості роботи діагностування	Держава зацікавлена у впровадженні систем автоматизації в заклади діагностики для прискорення та покращення роботи	Готовність до збільшення клієнтів, отримання відгуків про роботу продукту, його оновлення та покращення.

Продовження таблиці 4.7

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3	Зацікавленість компаній, що створюють апарати діагностування	Приватні та державні компанії зацікавленні у впровадженні процесу автоматизації роботи та пришвидшення процесу діагностування	Готовність до збільшення клієнтів, отримання відгуків про роботу продукту, його оновлення та покращення.
4	Розвиток та покращення технології	Приєднання людей, що хочуть покращити методи діагностики хвороб	Отримання відгуків та рекомендацій щодо покращення структури алгоритму, впровадження їх

Потрібно також проаналізувати загальні риси конкуренції на ринку. Це є важливим щоб виявити всю необхідну інформацію про можливих конкурентів, товарів-замінників та товарів-аналогів. Результати ступеневого аналізу конкуренції на ринку наведено в таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - олігополія	Є конкурентні продукти, що мають схожий частково функціонал, проте він повністю не повторюється	Для початку розробити технологію, що видає швидкий та точний результат. Після впровадження клієнтам отримувати відгуки та вдосконалювати її.

Продовження таблиці 4.8

<b>Особливості конкурентного середовища</b>	<b>В чому проявляється дана характеристика</b>	<b>Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)</b>
2. За рівнем конкурентної боротьби міжнародний	Продукти конкурентів розроблялись інтернаціональними командами	Маркетинг за допомогою Інтернет ресурсів для охоплення великої кількості потенційних користувачів. Участь в міжнародних ІТ та медичних заходах.
3. За галузевою - внутрішньогалузева	Програмне забезпечення використовується у сфері медицини	Розробка всього необхідного функціоналу для отримання якісного продукту.
4. Конкуренція за видами товарів - товарно-родова	Конкуренція існує між різними методами розробки технології обробки даних	Розробка всього необхідного функціоналу, прискорення роботи програми, зниження імовірності помилки до 0%.
5. За характером конкурентних переваг нецінова	Конкуренція існує шляхом вдосконалення та впровадження нових методологій в обробку даних ЕКГ	Потрібно розробити якісний продукт та провести всі можливі тести на перевірку роботи та результатів.
6. За інтенсивністю - не марочна	Вплив бренду незначний, головним є робота ПЗ та надання точних результатів	Участь в конференціях, семінарах, різних ІТ та медичних заклад для просування технології.

Також потрібно зробити детальний аналіз конкуренції. За допомогою метода п'яти сил конкуренції Майкла Портера він проведений. Виконується пошук прямих та потенційних конкурентів, товарів-замінників, а також вплив їх на впровадження продукту на ринок. Визначаються фактори сили постачальників та споживачів. Результати наведені у таблиці 4.9

Таблиця 4.9 – Аналіз за методом п'яти сил конкуренції Майкла Портера

	<b>Прямі конкуренти в галузі</b>	<b>Потенційні конкуренти</b>	<b>Постачальники</b>	<b>Клієнти</b>	<b>Товари-замінники</b>
<b>Складові аналізу</b>	Cardiolund ECG Parser	Відсутні	Діагностичні центри	Проект орієнтований на діагностичні центри кардіології	Більший функціонал системи, допоміжні можливості
<b>Висновки</b>	На ринку вже є досить інтенсивна конкурентна боротьба	Потенційних конкурентів немає, є можливість входження на ринок	Постачальники не диктують умови роботи на ринку	Лікарі диктують умови роботи, а саме точність та швидкість роботи технології	Можуть бути більш привабливими для діагностичних центрів

Можна зробити висновок, що запуск проекту є можливим виходячи з проведеного аналізу конкуренції на ринку. Прямими конкурентами в галузі є проект Cardiolund ECG Parser, потенційні ж конкуренти відсутні. Конкуренція на ринку є та вона досить інтенсивна. Лише клієнти диктують умови роботи програми, а саме швидкість роботи та детальність результатів. Постачальниками даних для обробки є насамперед діагностичні центри. Наведемо обґрунтування головних



властивостей на основі отриманих раніше даних, які забезпечать проекту перевершеність перед конкурентами. Його підсумки наведені у таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Використання паттернів для порівняння даних	Використання методу лінгвістичного моделювання для перетворення даних в ланцюг із символів, створення шаблонів таких ланцюгів
2	Швидке виконання аналізу даних	Створення словників, використання з них інформації для діагностування, порівняння аномальних даних з ЕКГ пацієнта
3	Можливість оновлення результатів	Інформація в файлах з результатами може оновлюватись за рахунок обробки нових даних ЕКГ
4	Інтеграція з іншими програмними забезпеченнями	Можливість впровадження технологію для подальшого аналізу та класифікації даних

За даними, що вказані в таблиці 4.10 можна провести аналіз факторів конкурентоспроможності, їх оцінка та рейтинг товарів від конкурентів. Підсумки порівняння наведені у таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Використання паттернів для порівняння даних	20	X						

Продовження таблиці 4.11

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
2	Швидке виконання аналізу даних	18		X					
3	Можливість оновлення словників	16	X						
4	Інтеграція з іншими ПЗ	15							X

Фінальним етапом аналізу проекту порівняно з конкурентами є складання SWOT- аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities). Для цього необхідно просумувати сильні та слабкі сторони проекту, а також його можливості та загрози, які були отримані в попередніх дослідженнях. Результати підсумку показано у таблиці 4.12.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

<b>Сильні сторони:</b> <ul style="list-style-type: none"> <li>– використання методу лінгвістичного моделювання;</li> <li>– створення словників із лінгвістичних ланцюгів сигналів ЕКГ;</li> <li>– можливість інтеграції з іншими програмними забезпеченнями.</li> </ul>	<b>Слабкі сторони:</b> <ul style="list-style-type: none"> <li>– недостатня кількість пам'яті для збереження даних;</li> <li>– складність розробки.</li> </ul>
<b>Можливості:</b> <ul style="list-style-type: none"> <li>– впровадження систем автоматизації процесу діагностування в медичні центри;</li> <li>– пришвидшення роботи діагностування хвороб.</li> </ul>	<b>Загрози:</b> <ul style="list-style-type: none"> <li>– недостатня кваліфікація розробників;</li> <li>– продукти конкурентів;</li> <li>– недовіра використання автоматизованих систем;</li> <li>– недостатність коштів реалізації.</li> </ul>

Після того, як було проведено SWOT аналіз, зробимо дослідження альтернатив ринкового впровадження проекту в залежності від використаних технологій, строків реалізацій альтернатив та вирогідність отримання ресурсів від результатів. Дані можливі варіанти реалізації ідеї показано в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

<b>№ п/п</b>	<b>Альтернатива (орієнтовний комплекс заходів) ринкової поведінки</b>	<b>Імовірність отримання ресурсів</b>	<b>Строки реалізації</b>
1	Презентація продукту на ІТ заходах	Середня	6-12 місяців
2	Продаж прав на використання технології	Висока	6 місяців
3	Презентація технології на закладах, пов'язаних з медициною	Висока	6-12 місяців

Після аналізу альтернатив було обрано продаж права на використання технології ІТ компаніям чи компаніям, що розробляють системи діагностування.

#### 4.4 Розроблення ринкової стратегії проекту

Першим кроком для розробки ринкової стратегії потрібно визначити цільові групи споживачів продукту, хто буде його використовувати. Програма направлена на інтеграцію та впровадження в системи діагностування. Немає можливості його використання як самостійний продукт. Технологія використовується як попередній етап аналізу даних. Тому впровадження програми напряму в медичні заклади не є доцільним. Найвигіднішим споживачем є компанії, що розроблять повноцінні апарати діагностування або лише програмне забезпечення для роботи таких приладів. Визначення цільових потенційних клієнтів показано в таблиці 4.14.

Таблиця 4.14 – Визначення цільових груп споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приватні та державні компанії, що створюють апарати кардіології	Споживачі готові прийняти продукт	30%	Інтенсивна	Складний
2	ІТ компанії, що розробляють експертно-діагностичні системи	Споживачі готові прийняти продукт	80%	Інтенсивна	Просто
Які цільові групи обрано: Приватні та державні компанії, що створюють апарати для центрів кардіології					

Налагодження зв'язків зі споживачами, не залежно від цільової аудиторії, є важливим етапом для отримання прибутку та розвитку стартапу. Потрібно грамотно визначити всі можливі стратегії для реалізації продукції. Не залежно від того, яка з цільових груп буде обрана необхідно налагодити зв'язки з її представниками задля реалізації продукту, що може стати проблемою.

Продукт націлений на конкретний споживчий сегмент. Було обрано базову стратегію розвитку продукту для взаємодії зі споживачами. Проте слідування тільки вибраної стратегії може не дати бажаних результатів. Якщо неправильно визначити ризики, продукт може навіть не вийти на ринок. Також потрібно завчасно обрати альтернативу, щоб запобігти припиненню розвитку стартапу. Опції можна переглянути в таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвитку

<b>Обрана альтернатива розвитку проекту</b>	<b>Стратегія охоплення ринку</b>	<b>Ключові конкурентос-проможні позиції відповідно до обраної альтернативи</b>	<b>Базова стратегія розвитку</b>
Стратегія диференціації	Концентрації	Розширення функціональних вимог відповідно до потреб користувача	Спеціалізації

Стратегію спеціалізації було обрано як базову, бо основною характеристикою є налаштування роботи програми під окремий цільовий сегмент. У разі провалу даної альтернативною буде обрано стратегію диференціації – орієнтування на потреби користувача.

Стратегія конкурентної поведінки продукту є не менш важливою. Інформацію про обрану стратегію наведено у таблиці 4.16.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

<b>Чи є проект «пер-шопрхідцем» на ринку?</b>	<b>Чи буде компанія шукати нових спо-живачів, або заби-рати існуючих у конкурентів?</b>	<b>Чи буде компанія копіювати основні характеристики товару конкурента, і які?</b>	<b>Стратегія конку-рентної поведінки</b>
Ні	Забирати існуючих	Копіювання спільних етапів функціоналу	Стратегія заняття конкурентної ніші

Обраною стратегією конкурентної поведінки є заняття конкурентної ніші з орієнтацією на декілька сегментів. Головною діяльністю є підтримку клієнтів та

розширення функціоналу. Наведемо підсумки попередніх досліджень стратегій поведінки:

- базова стратегія розвитку – спеціалізація;
- альтернативна стратегія розвитку – диференціація;
- конкурентна поведінка – заняття конкурентної ніші;
- обрані сегменти ринку – державні і приватні компанії, ІТ компанії.

Необхідно обрати стратегію позиціонування створеного продукту. Це робиться на основі обраних сегментів споживачів, їх потреб, конкурентної поведінки та стратегії розвитку. Результати показано у таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціонування

<b>Вимоги до товару цільової аудиторії</b>	<b>Базова стратегія розвитку</b>	<b>Ключові конкуренто-спроможні позиції власного стартап-проекту</b>	<b>Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)</b>
Безпомилкова обробка даних ЕКГ з виданням детального результату	Спеціалізація	Точність розпізнавання аномалій та швидкість обробки відео	Попередня обробка сигналів електрокардіограми; символічне представлення сигналу; збереження результатів та апарату нечітких множин.

## 4.5 Розроблення маркетингової програми стартап-проекту

Останнім етапом в розробці стартап-проекту є створення маркетингової програми. Для початку необхідно описати вигоди та переваги, що отримує клієнт після використання програми.

Інформація про ключових переваг концепції потенційного товару визначена в таблиці 4.18.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Детальний вивід результату	Результат отримується у вигляді детального опису сигналу ЕКГ	Використання лінгвістичного методу обробки
2	Наявність збереження результатів	Можливість зберегти результати в файл та в подальшому зчитувати і використовувати для аналізу	Є можливість збереження результат в файл
3	Швидкість та точність аналізу результатів в алгоритмах класифікації	Видається детальний результат у вигляді символьного ряду	Використання лінгвістичного методу обробки

Надалі розробляється трирівнева маркетингова модель товару: формується уточнення ідеї продукту, його сутність та складові, особливості процесу його надання. Необхідно визначити яким є товар за задумом, яким він є у реальному

виконанні та перспективи його розробки. Опис трьох рівнів моделі товару наведено в таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення, що обробляє дані ЕКГ		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Детальний опис сигналу ЕКГ	М	Тх/Тл
	2. Візуалізації сигналу ЕКГ в символьному вигляді	М	Тх/Тл
	3. Змога зберегти результат в файл	М	Тх/Тл
	Програма пройшла тестування, ISO/IEC 9126-1 2013		
	Постачається в форматі електронної бібліотеки		
	Марка:ECK Parser		
III. Товар із підкріпленням		До продажу: Програмне забезпечення	
		Після продажу: Розширення функціоналу	
Захист від копіювання: патент на алгоритму попередньої обробки ЕКГ, законодавство про авторське право			

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання. Визначення цінових меж буде наступним кроком. Необхідно враховувати аналіз рівня доходів споживачів, рівень цін на товари конкурентів та аналогів при встановленні ціни на потенційний товар (таблиця 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
15 тисяч гривень	Невідомі	Середній клас населення	Від 15 тисяч гривень за ліцензію на використання



Потрібно також визначити специфіку закупівлі клієнтів даного продукту, оптимальну систему збуту розробленого продукту. Результати наведені у таблиці 4.21.

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Цільова аудиторія – компанії, які бажають розробляти якісні та швидкі системи діагностування, що будуть точно проводити аналіз даних та безпомилково виводити результат. впровадити у своїй роботі	Встановити контакт можливими із споживачами та підтримання їх, швидке реагування на проблеми	Від виробника до споживача	Використання прямого каналу збуту до споживача, виконати мінімізацію збутових витрат, розвиток маркетингового спілкування із споживачем.

Виходячи з результату таблиці 4.21 продукт буде надходити до споживача напряму без посередників. Тому важливим етапом є встановлення та підтримка контакту із споживачами.

Останнім та не менш важливим етапом розробки маркетингової програми є визначення комунікації. Інформація встановлюється виходячи з визначеної специфіки поведінки клієнтів, попередньо обрану основу для позиціонування. Враховуються канали комунікації зі споживачами та визначаються рекламні концепти. Результати наведені в таблиці 4.22.

Таблиця 4.22 - Концепція маркетингових комунікацій

<b>Специфіка поведінки цільових клієнтів</b>	<b>Канали комунікацій, якими користуються цільові клієнти</b>	<b>Ключові позиції, обрані для позиціонування</b>	<b>Завдання рекламного повідомлення</b>	<b>Концепція рекламного звернення</b>
Оцінка точності та швидкості роботи альтернатив	Інтернет ресурси, спеціалізовані заклади, виставки	Швидкість обробки та точність обробки даних	Показати клієнтам функціонал продукту. Підштовхнути до його придбання	Автоматизуй системи діагностування хвороб, досліджуй нові можливості

### Висновки до розділу

У даному розділі було визначено всі етапи розробки стартап-проекту програмного забезпечення попередньої обробки ЕКГ. А саме проведено аналіз ринку та обрано цільову аудиторію, яка могла б отримати вигоду від використання продукту.

Технології для розробки продукту наявні та є у вільному доступі, тому створення програми є можливим в короткі терміни часу. Єдиним фактором для затримки розробки є проходження сертифікації в медичній сфері.

Був проведений аналіз ринку, визначено плюси та мінуси програмного забезпечення порівняно з конкурентами. Визначено всі можливі стратегії для успішного запуску стартапу. Розрахована імовірність успішності впровадження проекту, прораховані ризики та можливості життя проекту після його реалізації.

Завдяки інтеграції його до інших систем та підтримки його роботи з цими системами можна зробити продукт комерційним. Через унікальність функціоналу є значні шанси на впровадження його на ринок.

## ВИСНОВКИ

Залучення сучасних технологій для розробки нових методів у лікуванні серцево-судинних захворювань та запобіганню можливих хвороб є дуже актуальною темою.

Основною метою даної роботи є збільшення потенціалу проведення аналізу даних електрокардіограм за рахунок використання лінгвістичної моделі та додаванням функції приналежності та правил формування нечітких множин. Призначення даної роботи є створення програмного забезпечення, що реалізує описаний метод та надає зручний інтерфейс для роботи користувача.

В процесі виконання представленої магістерської дисертації реалізовано такі завдання дисертаційного дослідження:

- створено функцію зчитування даних та модуль розбиття на інтервали сигнал ЕКГ;
- розроблений програмний модуль для реалізації методу перетворення сигналу ЕКГ в лінгвістичний ланцюг шляхом додавання описати функції приналежності та правил нечітких множин;
- створений програмний модуль для візуалізації та збереження результатів обробки;
- досліджено ефективність удосконаленого методу.

Реалізацію завдань виконано у вигляді програмного забезпечення, що має простий інтерфейс. Користувач має можливість вибрати вхідний файл, отримати лінгвістичний ланцюг кожного сегменту даних, його візуальне представлення та зберегти результати для подальшого використання.

Було проведено експеримент з використанням алфавіту з 52-ма символів. На виході отримано файл із даними про кожну нечітку множини певного символу та окремий файл із перетворення кожного сегменту вхідного сигналу в лінгвістичний ланцюг.

У дисертаційній роботі зображені результати, що представляють собою рішення задачі обробки даних ЕКГ в рамках поставленої мети. Проведені

експериментальні дослідження показали, що зарекомендований алгоритм може бути успішно використаний в експертно-діагностичних системах. При зворотному відтворенні даних в числовий ряд середнє значення похибки складає 4,17%.

Під час виконання етапу дослідження реалізовувалось порівняння результатів перетворення в лінгвістичний ланцюг за допомогою апарату нечітких множин та структурним методом. Виконувався алгоритм оберненого перетворення для обох методів та пошук відхилення результатів від початкових даних. Проаналізувавши результати можна зазначити, що удосконалений метод видає більш точні результати перетворення. Середнє значення похибки методу з апаратом нечітких множин на 1,6% менша ніж для структурного методу.

Отримані теоретичні результати можуть бути використані для інтелектуального аналізу даних і обробки медико-біологічної інформації. Середовище проектування: C#, платформи, на яких проводилися дослідження: Microsoft Windows.

Проведено аналіз власного продукту у вигляді стартап-проекту. Виконано розгляд перспективи торгової реалізації виробу, рівень його потреби у споживачів та його рентабельність. Зроблено аналіз ринкових перспектив і можливостей виходу продукту на ринок в умовах високої конкуренції. На ринку наявна монополістична та товарно-родова конкуренція. Існують декілька фірм, що надають схожу продукцію, але їх товар трохи відрізняється, тому вихід на ринок не буде легким і вимагає детальної розробки стратегії для впровадження та розширення стартапу. Для введення реалізації проекту на ринок важливо обрати базову стратегію та альтернативу у разі провалу першої. Продукт націлений лише на один споживчий сегмент, проте у разі складного виходу на ринок або непередбачуваних ризиків, можливо розширення властивостей, які будуть відмінними від товарів конкурентів.

Матеріали дисертації було опубліковано у вигляді тез доповіді на конференції [36].

## ПЕРЕЛІК ПОСИЛАНЬ

1. Sricharan V. , Balamurali M., Vignesh, Preejith, Jayaraj J., Mohansankar S. Interpreting Deep Neural Networks for Single-Lead ECG Arrhythmia Classification (2020). arXiv preprint arXiv:2004.05399
2. Rahul Kumar Sevakula, Wan-Tai M. Au-Yeung, Jagmeet P. Singh, E. Kevin Heist, Eric M. Isselbacher, Antonis A. Armoundas State-of-the-Art Machine Learning Techniques Aiming to Improve Patient Outcomes Pertaining to the Cardiovascular System. – Journal of the American Heart Association. 2020
3. Baklan, I., Oliinyk, Y., Mukha, I., Lishchuk, K., Gavrilenko, O., Ocheretianyi, O., Tsytsyliuk, A.. Adaptive Multistage Method of Anomalies Detection in ECG Time Series. *COLINS*. 2020
4. Sajad Mousavi, Fatemeh Afghah, Fatemeh Khadem, U. Rajendra Acharya. ECG Language Processing (ELP): a New Technique to Analyze ECG Signals(2020). arXiv preprint arXiv:2006.08841
5. Логвинчук А. І. Виявлення аномалій в часових рядах довільної природи : магістерська дис. : 126 Інформаційні системи та технології / Логвинчук Андрій Ігорович. - Київ, 2019. - 79 с.
6. Кульчицька І.О., Тимченко О.В. Особливості алгоритмів бінаризації зображень документів – НУ «Львівська політехніка», 2013 – 141-143 с.
7. Simon Jaxy. Teaching a Machine to Diagnose a Heart Disease (2020). arXiv preprint arXiv:2009.0107
8. Haddadi R., Abdelmounim E., El Hanine M., Belaguid A. Discrete Wavelet Transform Based Algorithm for Recognition of QRS Complexes (2014) arXiv preprint arXiv:1703.00075
9. E. B. Mazomenos, T. Chen, A. Acharyya, A. Bhattacharya, J. Rosengarten , K. Maharatna. A Time-Domain Morphology and Gradient based Algorithm for ECG Feature Extraction. (2009). Електронний ресурс - [https://eprints.soton.ac.uk/340407/1/paperKC-003107\\_mazomenos.pdf](https://eprints.soton.ac.uk/340407/1/paperKC-003107_mazomenos.pdf)

10. Zhanhong Zhou<sup>1</sup> , Xiaolong Zhai<sup>1</sup> , Chung Tin. Fully Automatic Electrocardiogram Classification System based on Generative Adversarial Network with Auxiliary Classifier (2020) arXiv preprint arXiv: 2004.04894
11. Seedahmed S. Mahmoud, Zahir M. Hussain, Irena Cosic, and Qiang Fang. Time-Frequency Analysis of Normal and Abnormal Biological Signals. (2015). School of Electrical and Computer Engineering, Australia.
12. Binhang Yuan, Wenhui Xing. Diagnosing Cardiac Abnormalities from 12-Lead Electrocardiograms Using Enhanced Deep Convolutional Neural Networks (2019) arXiv preprint arXiv: 1908.06802
13. Asim Darwaish, Farid Naït-Abdesselam, Ashfaq Khokhar. Detection and prediction of cardiac anomalies using wireless body sensors and Bayesian belief networks. (2019) arXiv preprint arXiv: 1904.07976
14. Mohammad Mahmudur Rahman Khan, Md. Abu Bakr Siddique, Shadman Sakib, Anas Aziz, Abyaz Kader Tanzeem, Ziad Hossain. Electrocardiogram Heartbeat Classification Using Convolutional Neural Networks for the Detection of Cardiac Arrhythmia. (2020) arXiv preprint arXiv: 2010.04086
15. Dongdong Zhang, Xiaohui Yuan, Ping Zhang. Interpretable Deep Learning for Automatic Diagnosis of 12-lead Electrocardiogram. (2020) arXiv preprint arXiv: 2010.10328
16. Jessica Lin, Eamonn Keogh, Stefano Lonardi, Bill Chui. A symbolic Representation of Time Series, with Implications for Streaming Algorithms – 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2013
17. Баклан І.В. Основі етапи побудови лінгвістичної моделі. – Вестник ХНТУ №2, 2013 – 43-45 с.
18. Markus Thill<sup>1</sup> , Sina Däubener, Wolfgang Konen<sup>1</sup> , Thomas Bäck. Anomaly Detection in Electrocardiogram Readings with Stacked LSTM Networks. (2019) Електронний ресурс - <http://ceur-ws.org/Vol-2473/paper10.pdf>
19. Rajni Rajni. Electrocardiogram Signal Analysis - An Overview. – International Journal of Computer Applications. 2013

20. Баклан І. В. Лінгвистичне моделювання: основи, методи, деякі прикладні аспекти / І. В. Баклан // Систем. технології.- 2011. - № 3. - с.10-19.
21. Lohvynchuk A., Baklan I. Linguistic approach for a time series anomaly detection – Slovak International Scientific Journal. – 2019. – №35, Vol. 1. – pp. 16-18.
22. Олізаренко С.А., Свмокіш А.В., Капранов В.О. Метод порівняння ступеню нечіткості між нечіткими множинами типу 1 та інтервальними нечіткими множинами. – Системи озброєння і військова техніка №2, 2018. – 136 с.
23. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений/ – М.: Мир, 1976. – 165 с
24. Michael J. Watts. Fuzzy Sets and Fuzzification (2007) [Електронний ресурс] : [Веб-сайт]. Режим доступу: <http://mike.watts.net.nz/>
25. N. Someyama. Some Inclusion Relations for Fuzzy Sets (2010). arXiv preprint arXiv: 2010.04758
26. Xinxing Wu, Guanrong Chen. Revisiting T-Norms for Type-2 Fuzzy Sets (2003). arXiv preprint arXiv: 2003.11953
27. Grauel A. Fuzzy Logic System for ECG Interpretation (2000)
28. Chung, F. L., & Lee, T. (1994). Fuzzy competitive learning. Neural Networks, 7(3), 539-551.
29. Bezdek, J. C. (2013). Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media.
30. Ali Adeli, Mendi Neshat. A Fuzzy Expert System for Heart Disease Diagnosis. – IMECS 2010.
31. “C# programming guide” [Електронний ресурс] : [Веб-сайт]. Режим доступу: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
32. “Getting Started with Windows Forms” [Електронний ресурс] : [Веб-сайт]. Режим доступу: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/getting-started-with-windows-forms?view=netframeworkdesktop-4.8>
33. “Windows Forms documentation” [Електронний ресурс] : [Веб-сайт]. Режим доступу: <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-5.0>

34. “Read Data from a CSV File in C#”. [Електронний ресурс] : [Веб-сайт].  
Режим доступу: <https://blog.testproject.io/2017/02/09/read-data-csv-file-in-c/>
35. “How to serialize and deserialize JSON in .NET”. [Електронний ресурс] :  
[Веб-сайт]. Режим доступу: <https://docs.microsoft.com/en-us/dotnet/standard/serialization/system-text-json-how-to?pivots=dotnet-5-0>
36. Олійник А.О., Баклан І.В. Використання лінгвістичної моделі для попередньої обробки даних ЕКГ / А.О. Олійник, І.В. Баклан / IV Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020 р.



## **ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ**

**Плакат 1 Загальна схема алгоритму перетворення даних в лінгвістичний ланцюг**

## **Плакат 2 Схеми алгоритму сегментації**

### **Плакат 3 Діаграма компонентів**

## **Плакат 4 Результати дослідження ефективності**

## ДОДАТОК Б ПРОГРАМНИЙ КОД

Клас ECGSample

```
public class ECGSample
{
    public int Index { get; set; }
    public double Value { get; set; }

    public ECGSample(int ind, double val)
    {
        Index = ind;
        Value = val;
    }
}
```

Клас FuzzySets

```
public class FuzzySets
{
    public int segmentId;
    public double startPoint, endPoint;
    public char symbol;
    public double memberFunc;
    public int elementCount;

    public FuzzySets(int id, double start, double end, char s, double func, int count)
    {
        segmentId = id;
        startPoint = start;
        endPoint = end;
        symbol = s;
        memberFunc = func;
        elementCount = count;
    }
}
```

Модуль зчитування вхідних даних з файлу CSV

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using LumenWorks.Framework.IO.Csv;
using CsvHelper;
```

```
using CsvReader = LumenWorks.Framework.IO.Csv.CsvReader;
```

```
namespace ECGPars
```

```
{
    public class CSVReader
    {
        private int _bufferSize;
        public string FileName { get; set; }
        public List<ECGSample> ECGSignalChunk;

        public CSVReader(string fileName)
        {
            FileName = fileName;
            ECGSignalChunk = new List<ECGSample>();
        }

        public void Read()
        {
            using (CsvReader csv =
                new CsvReader(new StreamReader(FileName), true))
            {
                csv.MissingFieldAction = MissingFieldAction.ReplaceByEmpty;
                int index = 0;
                CultureInfo culture = CultureInfo.InvariantCulture;
                int i = 0;
                while (csv.ReadNextRecord())
                {
                    double val = 0;
                    string data = csv[index, 0];
                    if (Double.TryParse(data, NumberStyles.Number, culture, out val))
                    {
                        ECGSignalChunk.Add(new ECGSample((int)csv.CurrentRecordIndex,
val));
                            i++;
                        }
                        index++;
                    }
                }
            }
        }
    }
}
```

### Модуль розбиття даних на сегменти

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace ECGPars
{
    public class Segmentation
    {
        public List<ECGSample> signal;
        public int countOfSegm;
        public int count;

        public Segmentation(List<ECGSample> input)
        {
            signal = input;
            countOfSegm = 0;
            count = signal.Count;
        }

        public void GetSegment()
        {
            double maxEcg = signal.Max(x => x.Value);
            List<ECGSample> RList = signal.Where(x => x.Value > maxEcg - 0.55 &&
x.Value <= maxEcg).ToList();
            ECGSample firstRItem = RList[0];
            Console.WriteLine($"count of R {RList.Count} max = {maxEcg}");

            List<int> fewR = new List<int>();
            fewR.Add(1);

            for (int i = 0; i < RList.Count - 1; i++)
            {
                if (RList[i + 1].Index - RList[i].Index <= 180)
                {
                    fewR[fewR.Count - 1]++;
                }
                else
                {
                    fewR.Add(1);
                }
            }

            int index = -1;

```



```

List<int> rPeak = new List<int>();
List<int> diff = new List<int>();
int first = 0;
int second = 0;

for (int i = 0; i < fewR.Count; i++)
{
    index += (int) Math.Ceiling((double) fewR[i] / 2);
    rPeak.Add(RList[index].Index);
    index += fewR[i] / 2;
}
for(int i = 0; i < rPeak.Count - 1; i++)
{
    diff.Add(rPeak[i+1] - rPeak[i]);
}
diff.Add(rPeak[rPeak.Count - 1] - rPeak[rPeak.Count - 2]);

}
}
}

```

#### Модуль перетворення даних

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace ECGPars
{
    public class ToCharChain
    {
        private char[] symbols;
        private double min, max;
        private List<ECGSample> _sample;

        public char[] chain;
        public List<FuzzySets> sets;

        public ToCharChain(List<ECGSample> sample)
        {
            _sample = sample;
            symbols =
"abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ".ToCharArray(
);
            chain = new char[_sample.Count];

```

```

    sets = new List<FuzzySets>();
}

public void CreateDictionary()
{
    int count = _sample.Count;

    List<ECGSample> ecgSort = _sample.OrderBy(x => x.Value).ToList();

    double s = 0f;
    int elem = 0;
    double last = 0f;
    double step = (double)count / symbols.Length;
    int countElem = 0;
    double first = ecgSort[0].Value;

    foreach (var e in ecgSort)
    {
        if (s >= step)
        {
            sets.Add(new FuzzySets(elem, first, last, symbols[elem], 1, countElem));
            s -= step;
            elem++;
            first = e.Value;
            countElem = 0;
        }
        last = e.Value;

        chain[e.Index] = symbols[elem];
        s++;
        countElem++;
    }

    if (sets.Find(x => x.symbol == symbols[elem]) == null)
    {
        sets.Add(new FuzzySets(elem, first, last, symbols[elem], 1, countElem));
    }
}

public void ChangeSets()
{

```

```

double x = 0f;
for (int i = 0; i < sets.Count-1; i++)
{
    x = (sets[i].endPoint + sets[i + 1].startPoint) / 2;
    sets[i].endPoint = x;
    sets[i + 1].startPoint = x;
}
}
}

```

### Модуль збереження результатів

```

using System.Collections.Generic;
using System.IO;
using System.Linq;
using Newtonsoft.Json;

namespace ECGPars
{
    public class WriteJson
    {
        public List<FuzzySets> sets;
        public List<FuzzySets> readSets;

        public WriteJson(List<FuzzySets> fuzzySets)
        {
            sets = fuzzySets;
        }

        public void WriteToJsonFile(string fileName)
        {
            File.WriteAllText(fileName, JsonConvert.SerializeObject(sets));
        }

        public void ReadJson(string fileName)
        {
            if (File.Exists(fileName))
            {
                string text = File.ReadAllText(fileName);
                var readS = JsonConvert.DeserializeObject<IEnumerable<FuzzySets>>(text);
                readSets = readS.ToList();
            }
        }
    }
}

```

```

    }
}
}
}

```

Модуль дослідження ефективності

Клас оберненого перетворення

```

using System;
using System.Collections.Generic;

namespace ECGPars
{
    public class InverseProcess
    {
        public char[] chain;

        public List<ECGSample> preprocess;
        public List<FuzzySets> fuzzySetses;

        public InverseProcess(char[] result, List<FuzzySets> readSet)
        {
            chain = result;
            fuzzySetses = readSet;
            preprocess = new List<ECGSample>();
        }

        public void GetInverse()
        {
            Dictionary<char, double> dict = new Dictionary<char, double>();

            Console.WriteLine($"\\n\\nGetInverse Fuzzy sets");
            foreach (var s in fuzzySetses)
            {
                dict.Add(s.symbol, (s.endPoint + s.startPoint) / 2);
            }

            for (int i = 0; i < chain.Length; i++)
            {
                preprocess.Add(new ECGSample(i, dict[chain[i]]));
            }
        }
    }
}

```

```

    }
}

```

Клас підрахування середньоквадратичної похибки

```

using System;
using System.Collections.Generic;

namespace ECGPars
{
    public class Deviation
    {
        public List<ECGSample> origin;
        public List<ECGSample> preprocess;

        public Deviation(List<ECGSample> orig, List<ECGSample> prepr)
        {
            origin = orig;
            preprocess = prepr;
        }

        public double GetDevaiton()
        {
            int count = origin.Count;
            double sum = 0f;

            for (int i = 0; i < count; i++)
            {
                sum += Math.Pow(preprocess[i].Value - origin[i].Value, 2);
            }

            return Math.Sqrt(sum / count);
        }
    }
}

```

Клас перетворення за допомогою структурного методу

```

using System;
using System.Collections.Generic;
using System.Linq;

namespace ECGPars
{
    public class ConvertByStatic

```

```

{
    private char[] symbols;
    private double min, max;
    private List<ECGSample> _sample;
    public char[] chain;

    public Dictionary<char, Tuple<double, double>> dict;

    public ConvertByStatic(List<ECGSample> sample)
    {
        _sample = sample;
        symbols =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ".ToCharArray(
);
        chain = new char[_sample.Count];
        dict = new Dictionary<char, Tuple<double, double>>();

    }

    public void CreateStatic()
    {
        int count = _sample.Count;
        min = _sample.Min(value => value.Value);
        max = _sample.Max(value => value.Value);

        double step = (max - min) / symbols.Length;
        double first = min;

        for (int i = 0; i < symbols.Length - 1; i++)
        {
            dict.Add(symbols[i], new Tuple<double, double>(first, first + step));
            first = first + step;
        }

        dict.Add(symbols[symbols.Length - 1], new Tuple<double, double>(first,
max));
    }

    public void GetConvert()
    {
        foreach (var ecg in _sample)
        {
            foreach (var key in dict.Keys)
            {

```



```

    }
}

```

### Модуль графічного інтерфейсу

```

using System;
using System.Collections.Generic;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using EcgParser;

namespace ECGParserGUI
{
    public partial class Form1 : Form
    {
        List<EcgSample> ecgSamples = new List<EcgSample>();
        private CSVReader reader;
        private List<FuzzySets> readJson;
        public Form1()
        {
            InitializeComponent();
            InitializeGraphs(chart_EcgSignal, -0.5, 0.8);
        }

        private void button_GetFile_Click(object sender, EventArgs e)
        {
            OpenFileDialog folder = new OpenFileDialog();
            if (folder.ShowDialog() == DialogResult.OK)
            {
                string openFile = folder.FileName;
                textBox_FileName.Text = openFile;
            }
        }

        private void InitializeGraphs(Chart chart, double yMin, double yMax)
        {
            chart.Legends.Clear();

            var series = chart.Series[0];
            if (chart.Series.Count > 1)
                chart.Series[1].ChartType = SeriesChartType.Point;
            var chartArea = chart.ChartAreas[0];
            chartArea.AxisY.Title = "Symbols";
        }
    }
}

```



```

    }

    private void PlotRaw(TaskScheduler uiScheduler)
    {
        WriteJson fileJson = new WriteJson();
        fileJson.ReadJson();
        readJson = fileJson.readSets;
        chart_EcgSignal.ChartAreas[0].AxisX.Maximum = 10;
        double interval = 0f;
        foreach (var read in readJson)
        {
            FuzzySets set = new FuzzySets(read.segmentId, read.startPoint, read.endPoint,
            read.symbol, read.memberFunc, read.elementCount);
            Task.Factory.StartNew(() =>
            {
                interval = Math.Round(Math.Abs(set.endPoint - set.startPoint), 5);
                chart_EcgSignal.ChartAreas[0].AxisY.Interval = interval;
                label_ChainResult.Text += interval;
            }, CancellationToken.None, TaskCreationOptions.None, uiScheduler);
        }
    }

    private void button_StartConvert_Click(object sender, EventArgs e)
    {
        if (textBox_FileName.Text != String.Empty)
        {
            MessageBox.Show("Завантажте файл", "Пустий файл",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            reader = new CSVReader(textBox_FileName.Text);
            ecgSamples = reader.ECGSignalChunk;

            TaskScheduler uiScheduler =
            TaskScheduler.FromCurrentSynchronizationContext();
            Task.Factory.StartNew(() => PlotRaw(uiScheduler));
        }
    }
}

```



Ім'я користувача:  
Попенко Володимир Дмитрович

Дата перевірки:  
14.12.2020 00:12:40 EET

Дата звіту:  
14.12.2020 01:13:05 EET

ID перевірки:  
1005447534

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
77149

Назва документа: Olijnik\_magistr\_ip92mp

Кількість сторінок: 72 Кількість слів: 12349 Кількість символів: 94053 Розмір файлу: 1.17 MB ID файлу: 1005737890

## 15.7% Схожість

Найбільша схожість: 3.81% з джерелом з Бібліотеки (ID файлу: 1000756884)

11% Джерела з Інтернету

232

Сторінка 74

13.5% Джерела з Бібліотеки

708

Сторінка 80

## 1.17% Цитат

Цитати

11

Сторінка 81

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

31